Fast and Accurate Unlexicalized Parsing via Structural Annotations

Maximilian Schlund, Michael Luttenberger, and Javier Esparza
Institut für Informatik
Technische Universität München
Boltzmannstraße 3
D-85748 Garching
{schlund,luttenbe,esparza}@model.in.tum.de

Abstract

We suggest a new annotation scheme for unlexicalized PCFGs that is inspired by formal language theory and only depends on the structure of the parse trees. We evaluate this scheme on the TüBa-D/Z treebank w.r.t. several metrics and show that it improves both parsing accuracy and parsing speed considerably. We also show that our strategy can be fruitfully combined with known ones like parent annotation to achieve accuracies of over 90% labeled F_1 and leaf-ancestor score. Despite increasing the size of the grammar, our annotation allows for parsing more than twice as fast as the PCFG baseline.

1 Introduction

As shown by (Klein and Manning, 2003), unlexicalized PCFGs can achieve high parsing accuracies when training trees are annotated with additional information. An annotation basically amounts to splitting each nonterminal into several subcategories, which can even be derived automatically (Petrov et al., 2006; Petrov and Klein, 2007). Currently used annotation strategies, e.g. parent annotation (Johnson, 1998) or selectively splitting special nonterminals (e.g. marking relative clauses) as in (Schiehlen, 2004), are mostly linguistically motivated (with the exception of the above mentioned automatic approach).

In this paper we study new heuristics motivated by formal language theory for improving the parsing accuracy of unlexicalized PCFGs by means of refining the nonterminals of the grammar: One heuristic splits a nonterminal X into a family of nonterminals $(X_d)_{d \in D}$ based on the notion of the *dimension* (also *Horton-Strahler number*) of a tree (Strahler, 1952; Esparza et al., 2007; Esparza et al., 2014).

The *dimension* of a rooted tree t is defined as the height of the highest perfect binary tree[1] we can obtain from t by pruning subtrees and contracting edges.[2]

A result of (Flajolet et al., 1979) shows that the dimension characterizes the *minimal* amount of memory that is required to traverse a tree. So, intuitively, parse trees of high dimension should indicate an unnaturally complex sentence structure requiring the reader to remember too many incomplete dependent clauses in the course of reading the sentence. Section 2 corroborates experimentally that, indeed, parse trees of natural language have small dimension.

Since dimension is a meaningful measure of complexity and parse trees have low dimension, we conjectured that annotating nonterminals with the dimension of the subtree rooted at them could improve parsing accuracy (see Fig. 1 for an illustration). Section 5 shows that this is indeed the case: The combination of the dimension annotation and the well known parent annotation technique leads to absolute improvements of more than 5% F_1, 7–8% leaf-ancestor score, and a relative reduction of the number of crossing brackets of over 25% compared to a plain PCFG baseline. At the same time, quite surprisingly, parsing speed more than doubles.

It could be argued that any other graph theoretical measure for the complexity of a tree could lead to similar results. For this reason we have also considered annotating nonterminals with the height of the subtree rooted at them (the height is the most basic measure related to trees). Our experiments show that height annotation is also beneficial but further refinement via parent annotation yields less improvements than for the dimension annotation.

[1] A binary tree of height h is *perfect* if it has 2^h leaves.

[2] In other words, the dimension of t is the height of the highest perfect binary tree which is a minor of t.

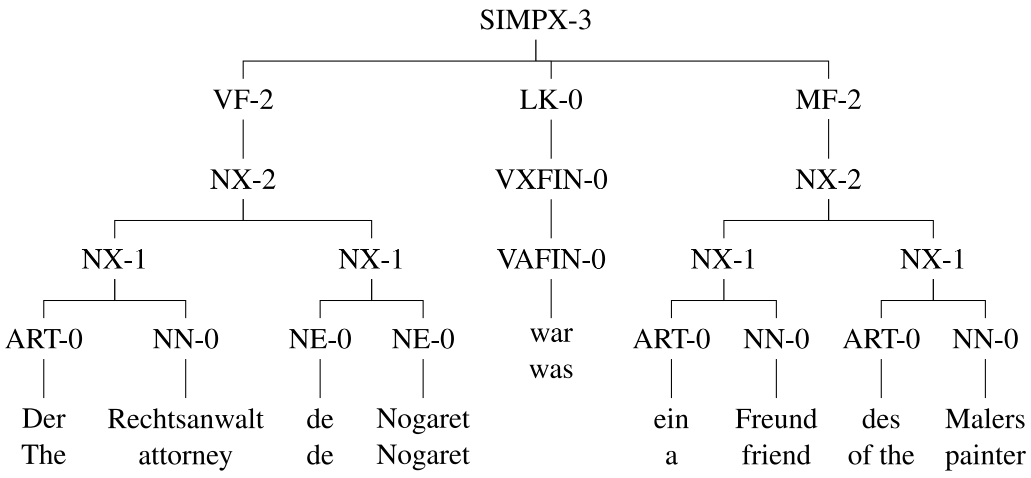

Figure 1: Dimension annotation of a tree from TüBa-D/Z: the label of every nonterminal is decorated with the dimension of the subtree rooted at it. The dimension of a parent node is the maximum of the dimensions of its children (*plus one* if this maximum is attained at least twice).

In the following two sections, we present more details on the use of tree dimension in NLP, continue with describing our experiments (Section 4) together with their results (Section 5), and finally conclude with some ideas for further improvements.

2 Tree Dimension of Natural Languages

We were able to validate our conjecture that parse trees of natural language should typically have small dimension on several treebanks for a variety of languages (cf. Table 1). The average dimension of parse trees varies only from 1.7 to 2.4 over all languages and the maximum dimension we ever encountered is 4.

3 Annotation Methods

In this paper we compare three different annotation methods: dimension, height, and parent annotation. The dimension (resp. height) annotation refine a given nonterminal X by *annotating* it with the dimension (resp. height) of the subtree rooted at it. A standard technique in unlexicalized parsing we compare against is *vertical markovization*, i.e. to refine nonterminals by annotating them with their parent (or grandparent) nonterminal (Klein and Manning, 2003).

Let us remark that we focus here only on methods to split nonterminals and leave merging strategies for further investigations. Amongst them *horizontal markovization* (Klein and Manning, 2003) is especially valuable for battling sparsity and can lead to more compact and often more accurate PCFGs.

Language	Average	Maximum
Basque	2.12	3
English	2.38	4
French	2.29	4
German(1)	1.94	4
German(2)	2.13	4
Hebrew	2.44	4
Hungarian	2.11	4
Korean	2.18	4
Polish	1.68	3
Swedish	1.83	4

Table 1: Average and maximum dimension for several treebanks of natural languages. Sources: English – 10% sample from the Penn treebank shipped with python nltk (Loper and Bird, 2002), German(2) – release 8 of the TüBa-D/Z treebank (Telljohann et al., 2003), the remaining treebanks are taken from the SPMRL shared task dataset (Seddah et al., 2013).

4 Methodology

4.1 Experimental Setup

We use release 8 of the TüBa-D/Z treebank (Telljohann et al., 2003) as dataset. To combine easy prototyping and data exploration with efficient parsing and standard evaluation methods we used python nltk (Loper and Bird, 2002) together with the Stanford parser (Klein and Manning, 2003). For evaluation we used the built in evalb, leaf-ancestor, and crossing brackets metrics provided by the Stanford parser. Is is important to note that all our experiments use gold tags from the treebank[3] which had the pleasant side effect that no parse failures were encountered. All experiments were carried out on a machine with an Intel i7 2.7 GHz CPU and 8 GB RAM and took about one week to run[4]. Our scripts and raw data can be obtained freely from `https://github.com/mschlund/nlp-newton`.

4.2 Randomization

We decided to sample our training- and test-data randomly from the treebank several times independently for each annotation strategy under test. This enables us to give more precise estimations of parsing accuracy (Section 5) and to assess their variability (cf. Figure 2). For each sample size N from $\{5k, 10k, 20k, \ldots, 70k\}$ we selected a random sample of size N from the set of all 75408 trees in the treebank. The first 90% of this sample was used as training set and the remaining 10% as test set. We then evaluated each of our six annotation methods on this same training/test set. The whole process was repeated ten times each, yielding 480 experiments altogether. For each experiment we evaluated parsing accuracy according to three evaluation measures as well as the parsing speed and the size of the derived grammar. Each of these numbers was then averaged over the ten random trials. To ensure perfect reproducibility we saved the seeds we used to seed the random generator.

4.3 Evaluation Measures

To thoroughly assess the performance of our annotation schemes we not only report the usual constituency measures (labeled precision/recall/F_1 and crossing brackets) proposed originally by (Abney et al., 1991) but also calculate leaf-ancestor scores (LA) proposed by (Sampson, 2000) since it has been argued that LA-scores describe the informal notion of a “good” parse better than the usual constituency measures. This is especially relevant for comparing parsing accuracy over different treebanks (Rehbein and Van Genabith, 2007a; Rehbein and van Genabith, 2007b).

5 Results

Our results are collected in Table 5. We measured a baseline accuracy of 84.8% labeled F_1-score for a plain PCFG without any annotations, lower than the 88% reported by (Rafferty and Manning, 2008) on a previous release of the TüBa-D/Z treebank (comprising only $20k$ sentences of length at most 40). However, the absolute improvements we found using annotations are consistent with their work, e.g. our experiments show an absolute increase of 3.4% when using parent annotation while (Rafferty and Manning, 2008) report a 3.1% increase. We suspect that the differences are largely suspect to the different data: considering sentences up to length 40, our experiments yield scores that are 1% higher. To explain all remaining differences we plan to replicate their setup.

5.1 Impact of Annotations

All three annotation methods (w.r.t. parent, dimension, height which we will abbreviate by PA, DA, HA for convenience) lead to comparable improvements w.r.t. constituency measures with small advantages for the two structural annotations. LA-evaluation on the other hand shows that HA and DA have a clear advantage of 3% over PA.

Quite surprisingly, both DA and HA can be fruitfully combined with parent annotation improving F_1 further by almost 2% and LA-metrics by 1–2% as well. However, the height+parent combination cannot compete with the dimension+parent method. One reason for this might be the significant increase in grammar size and resulting data-sparseness problems, although our learning curves (cf. Figure 2) suggest that lack of training data is not an issue.

Altogether, the DA+PA combination is the most precise one w.r.t. all metrics. It provides absolute increases of 5.6% labeled F_1 and 7.4–8.4% LA-score and offers a relative reduction of crossing brackets by 27%. This is especially relevant since according to (Manning and Schütze, 1999) a high number of crossing brackets is often considered “particularly dire”. Finally, this combination leads to a 60% increase in the number of exactly parsed sentences, significantly more than for the other methods.

5.2 Parsing Speed

We further study to what extent the three heuristics increase the size of the grammar and the time

[3] This is unrealistic of course, but is used for comparability with other work like (Rafferty and Manning, 2008).

[4] We only used a single core, since memory turned out to be the main bottleneck.

			evalb		Leaf-Ancestor		Crossing brackets	
Annotation	$\|G\|$	Speed ± stderr	F_1	exact	LA (s)	LA (c)	# CB	zero CB
Plain	21009	1.74 ± 0.04	84.8	24.4	84.0	79.7	1.17	58.5
Parent	34192	1.07 ± 0.01	88.2	31.8	86.6	82.9	1.07	61.8
Height	76096	3.06 ± 0.03	88.7	33.7	89.8	86.2	0.93	65.2
Height+parent	130827	2.20 ± 0.04	89.2	36.8	90.8	87.0	0.95	65.4
Dim	49798	$\mathbf{6.02} \pm 0.10$	88.5	31.8	89.7	86.1	0.90	64.9
Dim+parent	84947	4.04 ± 0.07	**90.4**	**39.1**	**91.4**	**88.1**	**0.85**	**67.2**

Table 2: Average grammar sizes, parsing speed, and parsing accuracies according to various metrics (for the $70k$ samples only, i.e. on 7000 test trees). All numbers are averaged over 10 independent random samples. $|G|$ denotes the number of rules in the grammar, parsing speed is measured in sentences per second. LA scores are reported as sentence-level (s) and corpus-level (c) averages, respectively. All accuracies reported in % (except # CB – the average number of crossing brackets per sentence).

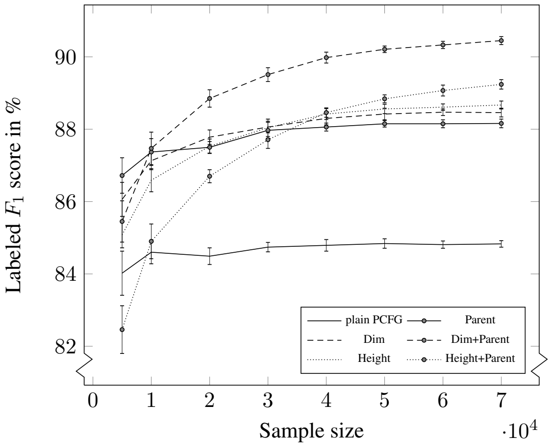

Figure 2: Learning curves for different annotation strategies. Average F_1 with standard deviation for random samples of various sizes (10 independent samples each).

needed to parse a sentence. As expected all three annotations increase the size of the grammar considerably (PA by 60%, DA by almost 140%, and HA by 260%). Surprisingly, our experiments did not show a direct influence of the grammar size on the average time needed to parse a tree: While parsing speed for PA drops by about 40%, DA and HA actually lead to significant *speedups* over the baseline (factor 3.4 for DA and 1.7 for HA). For the combination of dimension and parent annotation the gain in speed is less pronounced but still a factor of 2.3. One possible explanation is the fact that (for a grammar in CNF) a nonterminal of dimension d can only be produced either by combining one of dimension d with one of dimension strictly less than d or by two of dimension exactly $d - 1$. Since the dimensions involved are typically very small (cf. Table 1) this may restrict the search space significantly.

6 Discussion

We have described a new and simple yet effective annotation strategy to split nonterminals based on the purely graph-theoretic concept of *tree dimension*. We show that annotating nonterminals with either their dimension or their height gives accuracies that lie beyond parent annotation. Furthermore dimension and parent annotation in combination yield even higher accuracies (90.4% labeled F_1 and 91.4% LA-score on a sentence-level). Lastly, one of the most surprising findings is that, despite considerable growth of grammar size, parsing is significantly faster.

6.1 Future Work

We are currently experimenting with other treebanks like the SPMRL dataset (Seddah et al., 2013) which contains various "morphologically rich" languages (cf. Table 1). Although we cannot possibly expect to match the accuracies achieved by highly optimized lexicalized parsers with our simple annotation strategy alone, we are confident that our results transfer to other languages. A logical next step is to integrate our annotation methods into current parsing frameworks.

Since our annotations increase the size of the grammar significantly, horizontal markovization and more careful, selective dimension/height-splits (i.e. only carry out "profitable" splits) seem promising to avoid problems of data-sparsity – in particular if one wants to use further state-splitting techniques that are more linguistically motivated.

Finally, we are interested in understanding the parsing speedup incurred by dimension/height-annotations and to provide a theoretical analysis.

References

S. Abney, S. Flickenger, C. Gdaniec, C. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, M. Marcus, S. Roukos, B. Santorini, and T. Strzalkowski. 1991. Procedure for Quantitatively Comparing the Syntactic Coverage of English Grammars. In E. Black, editor, *Proceedings of the Workshop on Speech and Natural Language*, HLT '91, pages 306–311, Stroudsburg, PA, USA. Association for Computational Linguistics.

Javier Esparza, Stefan Kiefer, and Michael Luttenberger. 2007. An Extension of Newton's Method to ω-Continuous Semirings. In *Developments in Language Theory*, volume 4588 of *LNCS*, pages 157–168. Springer.

Javier Esparza, Michael Luttenberger, and Maximilian Schlund. 2014. A Brief History of Strahler Numbers. In *Language and Automata Theory and Applications*, volume 8370 of *Lecture Notes in Computer Science*, pages 1–13. Springer International Publishing.

Philippe Flajolet, Jean-Claude Raoult, and Jean Vuillemin. 1979. The Number of Registers Required for Evaluating Arithmetic Expressions. *Theoretical Computer Science*, 9:99–125.

Mark Johnson. 1998. PCFG Models of Linguistic Tree Representations. *Computational Linguistics*, 24(4):613–632.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 423–430, Stroudsburg, PA, USA. Association for Computational Linguistics.

Edward Loper and Steven Bird. 2002. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*, volume 999. MIT Press.

Slav Petrov and Dan Klein. 2007. Improved Inference for Unlexicalized Parsing. In *HLT-NAACL*, pages 404–411.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 433–440. Association for Computational Linguistics.

Anna N. Rafferty and Christopher D. Manning. 2008. Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines. In *Proceedings of the Workshop on Parsing German*, PaGe '08, pages 40–46, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ines Rehbein and Josef Van Genabith. 2007a. Evaluating Evaluation Measures. In *NODALIDA*, pages 372–379.

Ines Rehbein and Josef van Genabith. 2007b. Treebank Annotation Schemes and Parser Evaluation for German. In *EMNLP-CoNLL*, pages 630–639.

Geoffrey Sampson. 2000. A Proposal for Improving the Measurement of Parse Accuracy. *International Journal of Corpus Linguistics*, 5(1):53–68.

Michael Schiehlen. 2004. Annotation strategies for probabilistic parsing in german. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiorkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clérgerie. 2013. Overview of the SPMRL 2013 Shared Task: A Cross-Framework Evaluation of Parsing Morphologically Rich Languages. In *Proceedings of the 4th Workshop on Statistical Parsing of Morphologically Rich Languages: Shared Task*, Seattle, WA.

Arthur N. Strahler. 1952. Hypsometric (Area-Altitude) Analysis of Erosional Topology. *Bulletin of the Geological Society of America*, 63(11):1117–1142.

Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. 2003. Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z). Seminar für Sprachwissenschaft, Universität Tübingen, Germany.