Hybrid Ways to Improve Domain Independence in an ML Dependency Parser

Eckhard Bick
Institute of Language and Communication
University of Southern Denmark
5230 Odense M, Denmark
eckhard.bick@mail.dk

Abstract

This paper reports a hybridization experiment, where a baseline ML dependency parser, LingPars, was allowed access to Constraint Grammar analyses provided by a rule-based parser (EngGram) for the same data. Descriptive compatibility issues and their influence on performance are discussed. The hybrid system performed considerably better than its ML baseline, and proved more robust than the latter in the domain adaptation task, where it was the best-scoring system in the open class for the chemical test data, and the best overall system for the CHILDES test data.

1 Introduction

LingPars, a language-independent treebank-learner developed in the context of the CoNLL-X 2006 shared task (http://nextens.uvt.nl/~conll/), was inspired by the Constraint Grammar (CG) parsing approach (Karlsson et al. 1995) in the sense that it prioritized the identification of syntactic function over syntactic form, basing the dependency potential of a word on "edge" labels like subject, object etc. rather than the other way around. The system also used other features typical of CG systems, such as BARRIER conditions, tag chains of variable length, implicit clause boundaries and tag sets (Bick 2006). For the 2007 task only one such feature was newly introduced - a *directedness* marker for a few major functions, splitting subject, adverbial and adnominal labels into pairs of left- and right-attaching labels (e.g. SBJ-L, SBJ-R, NMOD-L, NMOD-R). Even this small addition, however, increased the memory space requirements of the model to such a degree that only runs with 50-75% of the training data were possible on the available hardware.

The main purpose of the LingPars architecture changes for CoNLL2007 (Nivre et al. 2007), however, was to test two core hypotheses:

- Can an independent, rule-based parser be made to conform to different, data-imposed descriptive conventions without too great a loss in accuracy?
- Does a rules-based dependency parser have a better chance than a machine-learned one to identify long-distance relations and global sentence structure, thus providing valuable arbiter information to the latter?

Obviously, both points rule out a test involving many languages with the *same* parser (CoNLL task 1). The domain adaptation task (task 2), however, satisfied the single-language condition and also adressed the descriptive adaptation problem (second hypothesis), involving three English treebanks - Wall Street Journal data from the Penn treebank (PTB, Marcus et al. 1993) for training, and the Pchem (Kulick et al. 2004) and CHILDES (Brown 1973 and MacWhinney 2000) treebanks with biomedical and spoken language data, respectively.

2 Developing and adapting EngGram

A parser with hand-written rules pays a high "labour price" to arrive at deep, linguistically pre-

Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pp. 1119–1123,
Prague, June 2007.

dictable and versatile analyses. For CG systems as employed by the author, the cost, from lexicon to dependency, is usually several man years, and results are not language-independent. One way of increasing development efficiency is to combine modules for different levels of analysis while reusing or adapting the less-language independent ones. Thus, the development of a new English dependency parser, EngGram, under way for some time, was accelerated for the present project by seeding the syntactic disambiguation grammar with *Danish* rules from the well-established DanGram parser (http://beta.visl.sdu.dk/ constraint_grammar.html). By maintaining an identical set of syntactic function tags, it was even possible to use the Danish dependency module (Bick 2005) with only minor adaptations (mainly concerning noun chains and proper nouns).

In order to integrate the output of a CG parser into an ML parser for the shared task data, several levels of compatibility issues have to be addressed. On the input side, (1) PTB tokenization and (2) word classes (PoS) have to be fed into the CG parser bypassing its own modules of morphological analysis and disambiguation. On the output side, (3) CG function categories and (4) attachment conventions have to be adapted to match PTB ones.

For example, the manual rules were tuned to a tokenization system that handles expressions such as "a=few", "at=least" and "such=as" as units. Though amounting to only 1% of running text, they constitute syntactically crucial words, and misanalysis leads to numerous secondary errors. Even worse is the case of the genitive-s (also with a frequency of 1%), tokenised in the PTB convention, but regarded a morpheme in EngGram. Since EngGram does not have a word class for the isolated 's', and since ordinary rules disfavour postnominal singel-word attachment, the 's' had to be fused in PTB-to-CG input, creating fewer tokens and thus problems in re-aligning the analysed output. Also relevant for a full structure parser is the parse window. Here, in order to match PTB window size, EngGram had to be forced not to regard ; ( ) and : as delimiters, with an arguable loss in annotation accuracy due to rules with global NOT contexts designed for smaller windows.

Finally, PTB convention fuses certain word classes, like subordinating conjunctions and prepositions (IN), and the infititive marker and the preposition "to" (TO). Though these cases can be treated by letting CG disambiguation override the CoNLL input's pos tag, input pos can then no longer be said to be "known", with some deterioration in recall as a consequence. Open class categories matched well even at a word-by-word level, closed class tokens were found to sometimes differ for individual words, an error source left largely unchecked.

Treebank error rate is another factor to be considered - in cases where the PoS accuracy of the human-revised treebanks is lower than that of a CG system, the latter should be allowed to *always* assign its own tags, rather than follow the supposedly fixed input pos. In the domain adaptation task, the CHILDES data were a case in point. A separate CG run indicated 6.6% differences in PoS, and manual inspection of part of the cases suggested that while some cases were irrelevant variations (e.g. adjective vs. participle), most were real error on the part of the treebank, and the parser was therefore set to ignore test data annotation and to treat it as pure text.

Errors appeared to be rarer in the training data, but inconsistencies between pos and function label (e.g. IN-preposition and SBJ-subject for "that") prove that errors aren't unknown here either - which is why a hybrid system with independent analysis has the potential benefit of compensating for "mis-learned" patterns in the ML system.

Output conversion from CG to PTB/CoNLL format had to address, besides realignment of tokens (e.g. genitive-s), the disparity in edge (function) labels. However, since the PTB set was more coarse grained, it was possible to simply lump several EngGram labels into one PTB label, for instance:

SC, OC, SUB, INFM --> VMOD
ADVL, SA, OA, PIV, PRED -> ADV

Some idiosyncrasies had to be observed here, for instance the treatment of SC (subject complement) as VMOD for words, but ADV for clauses, or the descriptive decision to tag direct objects in ACI constructions with OA-clausal complements as subjects. Some cases of label variation, however, could not be solved in a systematic way. Thus, adverbs within verb chains, always ADVL in EngGram, could not systematically be mapped, since PTB uses both VMOD and ADV in this position. A certain percentage of mismatches in spite of a correct analysis must therefore be taken into account as part of the "price" for letting the CG system advise the machine learner.

Dependencies were generally used in the same way in both systems, but multi word expressions were problematic, since PTB - without marking them as MWE - appears to attach all elements to a common head even where internal structure (e.g. a PP) is present. No reliable way was found to predict this behaviour from CG dependency output. Finally, PTB often uses the adverbial modifier tag (AMOD) for what would logically be the *head* of an expression:

*about **(head)** 1,200 (AMOD)*
*so **(head)** totally (AMOD)*
*herbicide **(head)** resistant (AMOD)*

EngGram in these examples regards the first element as AMOD modifier, and the second as head. Since the inversion was so common, it was accepted as either intentional or systematically erroneous, and the CG output inverted accordingly. It is an open question, for future research, whether the CG and ML systems could have been harmonized better, had the training data been an original dependency treebank rather than a constituent treebank, - or at least linguistically revised at the dependency level. Making the constituent-dependency conversion principles (Johansson & Nugues 2007, forthcoming) public *before* rather than after the shared task might also have contributed to a better CG annotation transfer.

3 System architecture

As described in (Bick 2006), the LingPars system uses the fine-grained part of speech (PoS) tags (POSTAG) and - for words above a certain frequency threshold - the LEMMA or, if absent, FORM tag. In a first round, LingPars calculates a preference list of functions and dependencies for each word, examining all possible mother-daughter pairs and n-grams in the sentence (or paragraph). Next, dependencies are adjusted for function, basically summing up the frequency-, distance- and direction-calibrated function->PoS attachment probabilities for all contextually allowed functions for a given word. Finally, dependency probabilities are weighted using linked probabilities for possible mother-, daughter- and sister-tags in a second pass.

The result are 2 arrays, one for possible daughter->mother pairs, one for word:function pairs. LingPars then attempts to "effectuate" the dependency (daughter->mother) array, starting with the - in normalized terms - highest value. If the daughter candidate is as yet unattached, and the dependency does not produce circularities or crossing branches, the corresponding part of the (ordered) word:function array is calibrated for the suggested dependency, and the top-ranking function chosen.

One of the major problems in the original system was uniqueness clashes, and as a special case, root attachment ambiguity, resulting from a conflict between the current best attachment candidate in the pipe and an earlier chosen attachment to the same head. Originally, the parser tried to resolve these conflicts by assigning penalties to the attachments in question and recalculating "second best" attachments for the tokens in question. While solving some cases, this method often timed out without finding a globally compatible solution.

In the new version of LingPars, with open resources, the attachment and function label rankings were calibrated using the analysis suggested by the EngGram CG system for the same data, assigning extra weights to readings supported by the rule based analysis, using addition of a weight constant for function, and multiplication with a weight constant for attachments, thus integrating CG information on par with statistical information[1]. This was

[1]Experiments suggested that there is a limit beyond which an increase of these weighting constants, for both function and dependency, will actually lead to a *decrease* in performance, because the positive effect of long-distance attachments from the CG system will be cancelled out by the negative effect of

not, however, thought sufficient to resolve the global syntactic problem of root attachment where (wrong) statistical preferences could be so strong that even 20 rounds of penalties could not weaken them sufficient to be ruled out. Therefore, root and root attachments supported by the CG trees were fixed in the first pass, without reruns. The same method was used for another source of global errors - coordination. Here, the probabilistic system had difficulties learning patterns, because a specific function label (SBJ or OBJ etc) would be associated with a non-specific word class (CC), and a non-specific function (COORD) with a host of different word classes. Again, adding a first-pass override based on CG-provided coordination links solved many of these cases.

Though limited to 2 types of global dependency (root and coordination), the help provided by the rule based analysis, also had indirect benefits by providing a better point of departure for other attachments, among other things because LingPars exaggerated both good and bad analyses: Good attachments would help weight other attachments through correct n-gram-, mother-, daughter- and sibling contexts, but isolated bad attachments would lead to even worse attachments by triggering, for instance, incorrect BARRIER or crossing branch constraints. These adverse effects were moderated by getting a larger percentage of global dependencies right in the first place, and also by a new addition to the crossing and BARRIER subroutine invalidating it in the case of CG-supported attachments.

4 Evaluation

The hybrid LingPars was the best-scoring system in the open section of both domain adaptation tasks[2] (Nivre et al. 2007), outperforming its probabilistic core system on all scores, with an improvement of 6.57 LAS percentage points for the pchemtb corpus (table 1), and 3.42 for the CHILDES attachment score (table 2). In the former, the effect was slightly more marked for attachment than for label accuracy.

However, whereas results also surpassed those of the top *closed class* system in the CHILDES domain (by 1.12 percentage points), they fell short of this mark for the pchemtb corpus - by 1.26 percentage points for label accuracy and 1.80 for attachment.

	Top score pchemtb	***average pchemtb***	***System pchemtb***	***System train***
Closed				
LAS	81.06	73.03	71.81	(75.01)[3]
UAS	83.42	76.42	74.71	(76.71)
LS	88.28	81.74	80.78	(84.12)
Open				
LAS	78.48	65.11	78.48	(79.04)
UAS	81.62	70.24	81.62	(80.82)
LS	87.02	77.14	87.02	(88.07)

Table 1: Performance, Pchemtb data

UAS	***Top score***	***average***	***System***
CHILDES closed	61.37	57.89	58.07
CHILDES open	62.49	56.12	62.49

Table 2: Performance, CHILDES data

When compared with runs on (unknown) data from the training domain, cross-domain performance of the closed system was 2 percentage points lower for attachment and 3.5 lower for label accuracy (LA scores of 71.81 and 58.07 for the pchemtb and CHILDES corpus, respectively).

Interestingly, hybrid results for the pchemtb data were only marginally lower than for the training domain (in fact, *higher* for attachment), suggesting a higher domain robustness for the hybrid than for the probabilistic approach.

disturbing the application of machine-learned local dependencies.

[2] During the test phase, the data set for one of the originally 2 test domains, CHILDES, was withdrawn from the official ranking, though its scores were still computed and admissible for evaluation.

[3] This is the accuracy for the test data used during development. For the PTB gold test data from track 1, LAS was higher (76.21).

References

E. Bick. 2006, LingPars, a Linguistically Inspired, Language-Independent Machine Learner for Dependency Treebanks, In: Màrquez, Lluís & Klein, Dan (eds.), *Proceedings of the Tenth Conference on Natural Language Learning (CoNLL-X, New York, June 8-9, 2006)*

E. Bick. 2005. Turning Constraint Grammar Data into Running Dependency Treebanks. In: Civit, Montserrat & Kübler, Sandra & Martí, Ma. Antònia (red.), *Proceedings of TLT 2005, Barcelona.* pp.19-2

R. Brown. 1973. A First Language: The Early Stages. Harvard University Press

R. Johansson and P. Nugues. 2007. Extended Constituent-to-Dependency Conversion for English. In: *Proceedings of NoDaLiDa 16.* Forthcoming

F. Karlsson, A. Vouitilainen, J. Heikkilä and A. Anttila. 1995. Constraint Grammar - A Language-Independent System for Parsing Unrestricted Text. Mouton de Gruyter: Berlin.

S. Kulick, A. Bies, M. Liberman, M. Mandel, R. McDonald, M. Palmer, A. Schein and L. Ungar. 2004. Integrated Annotation for biomedical Information Extractions. In: *Proceedings of HLT-NAACL 2004.*

B. MacWhinney. 2000. The CHILDES Project: Tools for Analyzing Talk. Lawrence Erlbaum

M. Marcus, B. Santorini and M. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. In: *Computational Linguistics Vol. 19,2.* pp. 313-330

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel and D. Yuret. 2007. The CoNLL 2007 Shared Task on Dependency Parsing. In: *Proceedings of the CoNLL 2007 Shared Task. Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL).*