Accurate Conversion of Dependency Parses: Targeting the Stanford Scheme

Katri Haverinen,[1] **Filip Ginter,**[1] **Sampo Pyysalo**[2] and **Tapio Salakoski**[1,2]
[1]Department of Information Technology
[2]Turku Centre for Computer Science (TUCS)
20014 University of Turku, Finland
first.last@utu.fi

Abstract

We present a conversion from the dependency scheme employed by the Pro3Gres parser to the Stanford scheme, as a further step towards unification of dependency schemes. An evaluation of the conversion shows that it is highly reliable, resulting in less than one percentage point performance penalty on the actual parser output. This supports the suitability of the Stanford scheme as a unifying representation and the applicability of our conversion formalism to parser scheme conversions. We further provide an evaluation of the Pro3Gres parser, thus adding it to the growing set of parsers evaluated under comparable conditions using the Stanford scheme.

1 Introduction

The development of parsing technologies has recently made it feasible to apply full parsers to many tasks where partial parsing was previously the approach of choice, such as information extraction (IE). In particular in biomedical IE, there has been substantial interest in the application of full dependency parsers in response to the relative complexity of the domain language and also due to the advantages of the immediate representation that dependency formalisms give to grammatical functions (e.g. *subject* and *object*).

Parsing technologies, however, differ substantially in the syntactic schemes employed. This has a number of unfortunate consequences: corpora tend to be formalism-specific, reducing the amount of data available, evaluations of parsers yield results that cannot be directly compared, and methods that apply parsers tend to become bound to a particular scheme. Both parser developers and those who apply parsers would benefit from a reduction of this fragmentation.

In this study, we consider a full dependency parser, Pro3Gres (Schneider et al., 2004), which has been developed with particular attention to the challenges of biomedical domain text and applied in numerous domain studies. Pro3Gres has been evaluated by its authors on a small dependency treebank in its native syntactic representation as well as in one of the CoNLL shared tasks on dependency parsing (Schneider et al., 2007); however, due to differences in syntactic representations it is difficult to directly relate these results to evaluations of other parsers in the domain. Here, we study the feasibility of translating the unique syntactic scheme of Pro3Gres into a more commonly used shared representation.

2 Related work

There has recently been a significant amount of work narrowing the gap between different parser output representations. Three prominent approaches are dependency-based: the Grammatical Relations (GR) dependency scheme, proposed by Carroll et al. (1998) for parser evaluation, the Stanford dependency scheme (SD) of de Marneffe et al. (2006), oriented towards applications such as IE, and the scheme that was introduced in the CoNLL shared dependency parsing tasks (Nivre et al., 2007). In this paper, we consider unification under the Stanford scheme.

The GR and SD schemes have been applied in a number of parser evaluation studies in which the native parser output was converted into the target dependency scheme. Table 1 summarizes estimated performance of the various conversions as

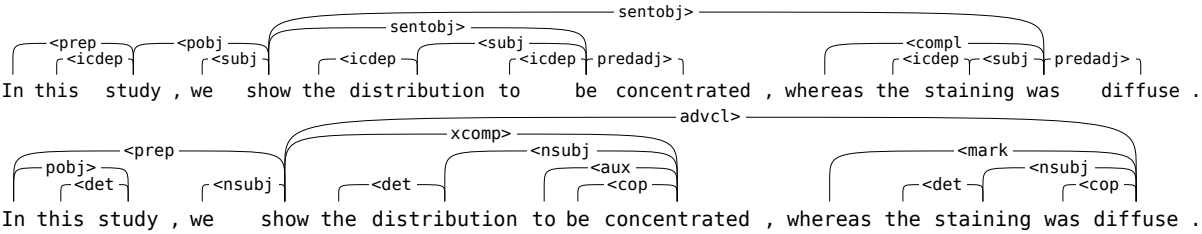

Figure 1: An example of the differences between the Pro3Gres scheme (top) and the Stanford scheme (bottom). Note the technical intra-chunk dependencies, *icdep*, in the Pro3Gres parse.

study	from	to	F
Clark and Curran (2007)	CCG	GR	84.8%
Pyysalo et al. (2007b)	LG	SD	97.1%
Sagae et al. (2008)	HPSG	GR	87.1%
Sagae et al. (2008)	SD	GR	74.5%
Sagae et al. (2008)	HPSG	PTB	98.1%

Table 1: Previously reported conversions with conversion quality estimates, given as F-scores.

reported by their authors.

There is a surprisingly large amount of variation in these results. While the results would appear to suggest that conversions into GR are particularly difficult, there are differences in conversion methodology that prevent clear conclusions from being drawn. Additionally, the schemes are different in the sense that some of them, including GR, are deep, whereas others are more surface-oriented. The development and evaluation of a conversion from the Pro3Gres native scheme to SD is thus an important point towards establishing whether highly accurate conversions into SD can be achieved in general.

3 Methods

We now briefly describe the Pro3Gres and SD schemes and the Pro3Gres→SD conversion. For details of the two schemes, see the papers by Schneider et al. (2004) and de Marneffe et al. (2006), respectively.

3.1 Pro3Gres parser and its dependency scheme

Pro3Gres is a dependency-based parser created by Schneider et al. (2004). A notable property of the parser is that it uses a chunker to extract noun and verb groups as a separate pre-parsing step.

The Pro3Gres scheme has a total of 23 dependency types, excluding the so called *intra-chunk dependencies* that are fully contained within chunks. As intra-chunk dependencies are not a primary output of the parser, and as they form a relatively flat structure, our conversion does not target them. However, in order to be able to recognize certain structures, such as passives, we introduce technical dependencies *icdep* from the chunk head to each token in the chunk. Figure 1 is an illustration of the Pro3Gres scheme as compared to the Stanford scheme.

3.2 Stanford dependency scheme

The Stanford dependency scheme (SD) is an application-oriented scheme introduced by de Marneffe et al. (2006). The scheme defines 48 dependency types that are arranged in a hierarchy. De Marneffe et al. also provide a method for converting parse trees from the PTB scheme into the SD scheme.

3.3 Pro3Gres→SD conversion

The Pro3Gres→SD conversion was carried out using 176 hand-written rules in the lp2lp dependency parse conversion formalism (see, e.g., Pyysalo et al. (2007b)).

One-to-one correspondences of dependency types are rare in the conversion. An example of a particularly difficult dependency type to translate is the Pro3Gres type *sentobj*. In SD, it corresponds to five different dependency types: *xcomp*, *partmod*, *infmod*, *ccomp* and *advcl*. In Figure 1 we illustrate two different uses of the *sentobj* type. Another issue that complicates the transformation rules is that some dependency types in SD, the most common example being the copula, cause substantial changes to the structure of the parse, as the head is chosen differently in the two schemes. This is, again, illustrated in Figure 1.

4 Results and discussion

We estimate the conversion performance in two separate ways: on an actual output of the

		gold standard	
		present	*absent*
system output	*present*	461	77 (73+4)
	absent	161 (156+5)	—

Table 2: Results of the manual analysis of the conversion quality. Parsing errors are divided between errors attributed to the Pro3Gres parser and errors attributed to the conversion. This division is shown in parentheses as *parser errors+conversion errors*. All numbers are dependency counts.

err.	P	R	F
incl.	85.7% (461/538)	74.1% (461/622)	79.5%
excl.	86.3% (461/534)	74.9% (466/622)	80.2%

Table 3: (P)recision, (R)ecall and F-score figures including and excluding conversion errors (based on the manual analysis reported in Table 2).

Pro3Gres parser and on a separate set of gold-standard Pro3Gres parses. The former evaluation is performed on the BioInfer corpus (Pyysalo et al., 2007a) which has gold-standard SD annotation. As performance measures, we use precision, recall, and F. The rules have been developed using 200 sentences from BioInfer as reference, we thus perform all BioInfer measurements on an evaluation set consisting of the remaining 900 sentences.

4.1 Evaluation of the Pro3Gres→SD conversion

To estimate the quality of the conversion, we manually analyse the converted Pro3Gres output on 30 sentences (622 dependencies) randomly drawn from the evaluation set of BioInfer sentences. We attribute each parsing error as caused either by the parser or by the conversion. The result of this analysis is presented in Table 2. We find that the conversion accounts for 4/77=5.2% of all precision errors and 5/161=3.1% of all recall errors. The conversion thus accounts for only a small percentage of the errors found in the converted parser output. In fact, the absolute penalty on the overall F-score of the parser is only 0.7 percentage points, as shown in Table 3.

The manual analysis estimates the performance of the rules on the actual parser output and is thus most relevant from the applied point of view and for parser evaluation. As seen in Table 3, Pro3Gres trades higher precision for lower recall. This often means that rare and exceptionally complex structures are not given any analysis. This, in turn, has the effect that also the conversion rules are not applied for these sections of the sentence and therefore cannot fail. In order to estimate the performance of the conversion in the ideal case of the parser producing a perfect analysis, we have annotated in both the Pro3Gres scheme and the SD scheme a set of 50 sentences (715 SD dependencies) randomly drawn from the GENIA corpus. On this set, we find that the conversion results in a 96.1% F-score (96.9% precision and 95.4% recall). The difference in conversion accuracy of the actual parser output as compared to the gold-standard output shows that as the parser coverage is increased in the future, corresponding conversion rules will need to be added.

4.2 Evaluation of the Pro3Gres parser

The Pro3Gres→SD conversion allows an evaluation of Pro3Gres performance on the SD-annotated BioInfer corpus, thus complementing the results previously reported by Clegg and Shepherd (2007) and Pyysalo et al. (2007b). This evaluation, however, is complicated by the fact that Pro3Gres chunks noun and verb groups and does not aim to generate sufficiently detailed chunk-internal analysis. To address this difference in resolution detail, we chunk the gold-standard data using the existing gold-standard annotation and only consider chunk-external dependencies in the evaluation (see Figure 2).

In Table 4 we report the performance of Pro3Gres on the 900 BioInfer evaluation sentences. The parser was used together with the GENIA tagger (Tsuruoka et al., 2005) and LTChunk chunker (Mikheev, 1997). As a point of comparison, we also report the performance of the Charniak-Lease parser (Lease and Charniak, 2005), a state-of-the-art, domain-adapted statistical parser. The Charniak-Lease output was transformed to the SD scheme using the Stanford conversion tools (de Marneffe et al., 2006). To assess the numerical comparability of the chunk-based evaluation strategy, we include the result reported by Pyysalo et al. (2007b) for the Charniak-Lease parser on full, unchunked BioInfer.

We observe that Pro3Gres achieves state-of-the-art performance, only slightly lower than that of the Charniak-Lease parser. Further, we note that the chunked evaluation strategy results in 3.5 percentage point performance penalty.

	Pro3Gres			Charniak-Lease			
chunked	P	R	F	P	R	F	ΔF
yes	78.5	70.5	74.3	74.4	77.5	75.9	1.6
no	-	-	-	78.4	79.9	79.4	-

Table 4: Performance of the Pro3Gres and Charniak-Lease parsers on the BioInfer corpus. The result for the Charniak-Lease parser on the unchunked BioInfer was reported by Pyysalo et al. (2007b).

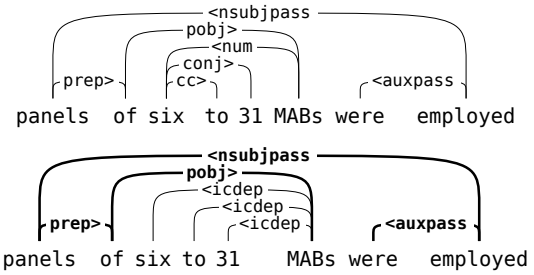

Figure 2: Original gold-standard structure (top) compared to chunked gold standard (bottom) with the intra-chunk structure flattened into *icdep* dependencies. The parser is only evaluated on the chunk-external dependencies, displayed in bold.

5 Conclusions

The main practical contribution of this paper is the set of rules for a very accurate conversion from the Pro3Gres scheme to the Stanford scheme (SD). In particular, on actual parser output, the conversion results in less than one percentage point penalty on the parser F-score performance. The conversion increases the applicability of Pro3Gres, as it enables it to produce output in a commonly used scheme.

Moreover, the ability to produce an accurate conversion into the SD scheme, already a third such conversion — the other two being the conversions from PTB (de Marneffe et al., 2006) and from LG (Pyysalo et al., 2007b) — suggests that the SD scheme does not pose significant problems as a conversion target. The SD scheme is also designed to be oriented towards applications, such as IE (de Marneffe et al., 2006). This study thus further strengthens the case for the adoption of the SD scheme as a unifying representation for full parsers in the applied domain, previously argued for by de Marneffe et al. (2006), Clegg and Shepherd (2007), and Pyysalo et al. (2007b).

The evaluation data, the conversion rules, and our modified version of the lp2lp implementation are available under an open-source license at `http://www.it.utu.fi/BioInfer`.

6 Acknowledgments

We thank Gerold Schneider for producing the Pro3Gres parses. This study was supported by the Academy of Finland.

References

J. E. Carroll, E. Briscoe, and A. Sanfilippo. 1998. Parser evaluation: a survey and a new proposal. In *Proc. of LREC'98*, pages 447–454.

S. Clark and J. Curran. 2007. Formalism-independent parser evaluation with CCG and DepBank. In *Proc. of ACL'07*, pages 248–255.

A. B. Clegg and A. Shepherd. 2007. Benchmarking natural-language parsers for biological applications using dependency graphs. *BMC Bioinformatics*, 8(1):24.

M. Lease and E. Charniak. 2005. Parsing biomedical literature. In *Proc. of IJCNLP'05*, pages 58–69.

M-C. de Marneffe, B. MacCartney, and C. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proc. of LREC'06*, pages 449–454.

A. Mikheev. 1997. Automatic rule induction for unknown-word guessing. *Computational Linguistics*, 23(3):405–423.

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 *Proc. of EMNLP-CoNLL'07*, pages 915–932.

S. Pyysalo, F. Ginter, J. Heimonen, J. Björne, J. Boberg, J. Järvinen, and T. Salakoski. 2007a. BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(50).

S. Pyysalo, F. Ginter, V. Laippala, K. Haverinen, J. Heimonen, and T. Salakoski. 2007b. On the unification of syntactic annotations under the Stanford dependency scheme: A case study on BioInfer and GENIA. In *Proc. of BioNLP'07*, pages 25–32.

K. Sagae, Y. Miyao, T. Matsuzaki, and J. Tsujii. 2008. Challenges in mapping of syntactic representations for framework-independent parser evaluation. In *Proc. of ICGL'08*.

G. Schneider, F. Rinaldi, and J. Dowdall. 2004. Fast, deep-linguistic statistical dependency parsing. In *Proc. of COLING'04 Recent Advances in Dependency Grammar*, pages 33–40.

G. Schneider, K. Kaljurand, F. Rinaldi, and T. Kuhn. 2007. Pro3Gres parser in the CoNLL domain adaptation shared task. In *Proc. of EMNLP-CoNLL'07*, pages 1161–1165.

Y. Tsuruoka, Y. Tateishi, J-D. Kim, T. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii. 2005. Developing a robust part-of-speech tagger for biomedical text. In *Proc. of PCI'05*, pages 382–392.