Influence of preprocessing on dependency syntax annotation: speed and agreement

Arne Skjærholt
Department of Informatics, University of Oslo
arnskj@ifi.uio.no

Abstract

When creating a new resource, preprocessing the source texts before annotation is both ubiquitous and obvious. How the preprocessing affects the annotation effort for various tasks is for the most part an open question, however. In this paper, we study the effects of preprocessing on the annotation of dependency corpora and how annotation speed varies as a function of the quality of three different parsers and compare with the speed obtained when starting from a least-processed baseline.

We also present preliminary results concerning the effects on agreement based on a small subset of sentences that have been doubly-annotated.[1]

1 Introduction

It is commonly accepted wisdom in treebanking that it is preferable to preprocess data before PoS and syntax annotation, rather than having annotators work from raw text. However, the impact of preprocessing is not well studied and factors such as the lower bound on performance for preprocessing to be useful and the return on investment of increased performance are largely unknown.

Corpora and applications based on dependency syntax have become increasingly popular in recent years, and many new corpora are being created. In this work we investigate the task of syntactic annotation based on dependency grammar, and how annotation speed and inter-annotator agreement are influenced by parser performance. Our study is performed in the context of the annotation effort currently under way at the national library of Norway, tasked with creating a freely available syntactically annotated corpus of Norwegian. It is the first widely available such corpus.

[1] Code and data used to obtain these results is available at https://github.com/arnsholt/law7-annotation

1.1 Related work

The Penn Treebank project (Marcus et al., 1993) had annotators correct automatically parsed and PoS-tagged data, and they report that correcting rather than annotating from scratch is massively helpful in the PoS annotation task (from scratch took twice as long and increased error rate and disagreement by 50%), but unfortunately there is no such comparison for the syntactic bracketing task. The task of PoS annotation has been studied further by Fort and Sagot (2010), who establish the lower bound on tagger accuracy to be in the range of 60–80% for the preprocessing to be useful.

For the task of syntactic bracketing, Chiou et al. (2001) investigated some facets of the problem while developing the Penn Chinese treebank and found that when using a parser with a labelled $F_1 = 76.04$, the time spent correcting is 58% of the time spent on unassisted annotation, and a further improved parser ($F_1 = 82.14$) reduces the time to 50% of that used by unassisted annotation.

2 Experimental protocol

In this section we outline the key methodological choices made for our experiments. First we discuss what timing data we collect and the texts annotated, before describing the preprocessors used.

Environment For our experiments, four different texts were chosen for annotation: two from the *Aftenposten* (AP 06 & AP 08), and two from *Dagbladet* (DB 12 & DB 13), both daily newspapers. Key statistics for the four texts are given in Table 1. The annotation effort uses the TrEd tool[2], originally created for the Prague Dependency Treebank project. It is easily extended, and thus we used these facilities to collect the timing data. To minimise interference with the annotators, we simply recorded the time a sentence was shown on screen and accounted for outliers caused by breaks and interruptions in the analysis.

The annotation work is done by two annotators, Odin and Thor. Both are trained linguists, and

[2] http://ufal.mff.cuni.cz/tred/

Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse, pages 28–32,
Sofia, Bulgaria, August 8-9, 2013.

Text	n	μ	s
AP 06	373	17.0	10.8
AP 08	525	16.5	9.11
DB 12	808	12.1	8.47
DB 13	648	14.6	9.15
Total	2354	34223 tokens	

Table 1: Statistics of the annotated texts. n number of sentences, μ mean length, s length standard deviation.

are full-time employees of the National Library tasked with annotating the corpus. The only additional instruction given to the annotators in conjunction with the experiment was that they try to close the TrED program when they know that they were going away for a long time, in order to minimise the number of outliers. The actual annotation proceeded as normal according to the annotation guidelines[3]. Thor annotated AP 08 and DB 13, while Odin annotated AP 06 and DB 12 as well as the first 400 sentences of DB 13 for the purposes of measuring annotator agreement.

Preprocessing In our experiments, we consider three different statistical parsers as preprocessors and compare these to a minimally preprocessed baseline. Unfortunately, it was impossible to get timing data for completely unannotated data, as TrED requires its input to be a dependency tree. For this reason our minimal preprocessing, we call it the caterpillar strategy, is attaching each word to the previous word, labelled with the most frequent dependency relation.

Of the three statistical parsers, one is trained directly on already annotated Norwegian data released by the treebank project (version 0.2) and the other two are cross-lingual parsers trained on converted Swedish and Danish data using the techniques described in Skjærholt and Øvrelid (2012). In brief, this technique involves mapping the PoS and dependency relation tagsets of the source corpora into the corresponding tagsets of the target representation, and applying structural transformations to bring the syntactic analyses into as close a correspondence as possible with the target analyses. It was also shown that for languages as closely related as Norwegian, Danish and Swedish, not delexicalising, contrary to the standard procedure in cross-lingual parsing (Søgaard, 2011; Zeman and Resnik, 2008), yields a non-negligible boost in performance.

[3]Distributed with the corpus at:
`http://www.nb.no/Tilbud/Forske/Spraakbanken/Tilgjengelege-ressursar/Tekstressursar`

Parser	UAS	LAS
Baseline	30.8%	3.86%
Danish	69.9%	46.7%
Swedish	77.7%	68.1%
Norwegian	86.6%	83.5%

Table 2: Parser performance. Labelled (LAS) and unlabelled (UAS) attachment scores.

All three parsers are trained using MaltParser (Nivre et al., 2007) using the liblinear learner and the nivreeager parsing algorithm with default settings. The Norwegian parser is trained on the first 90% of the version 0.2 release of the Norwegian dependency treebank with the remaining 10% held out for evaluation, while the cross-lingual parsers are trained on the training sets of Talbanken05 (Nivre et al., 2006) and the Danish Dependency Treebank (Kromann, 2003) as distributed for the CoNLL-X shared task. The parser trained on Swedish data is lexicalised, while the one trained on Danish used a delexicalised corpus.

The performance of the four different preprocessing strategies is summarised in Table 2. The numbers are mostly in line with those reported in Skjærholt and Øvrelid (2012), with a drop of a few percentage points in both LAS and UAS for all parsers, except for a gain of more than 5 points LAS for the Danish parser, due to the fixed relation labels. There are three reasons for the differences: First of all, the test corpus is different; Skjærholt and Øvrelid (2012) used the version 0.1 release of the Norwegian corpus, while we use version 0.2. Secondly, TrED requires that its input trees only have a single child of the root node, while MaltParser will attach unconnected subgraphs to the root node if the graph produced after consuming the whole input isn't connected. Finally, TrED validates dependency relation labels strictly, which revealed a few bugs in the conversion script for the Danish data. A post-processing script corrects the invalid relations and attaches multiple children of the root node to the most appropriate child of the root.

The texts given to the annotators were an amalgam of the outputs of the four parsers, such that each block of ten sentences comes from the same parser. Each chunk was randomly assigned

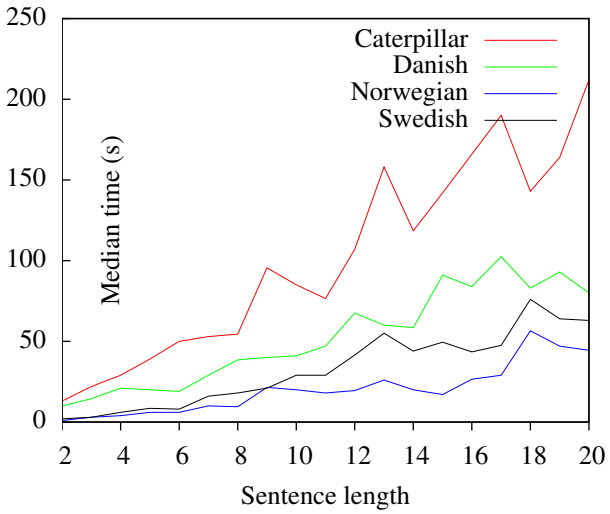

Figure 1: Median annotation time, Odin.

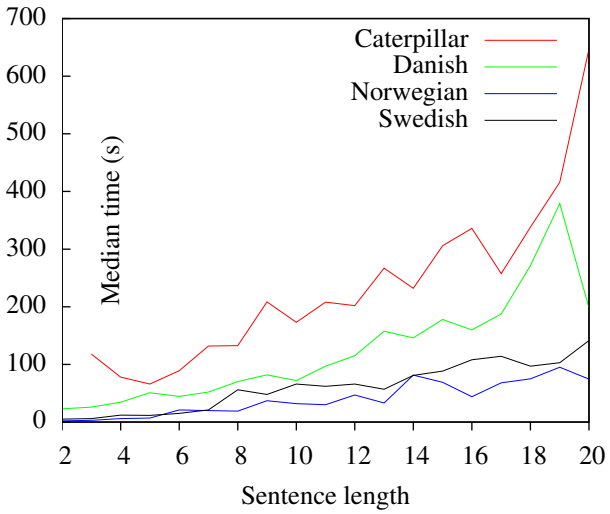

Figure 2: Median annotation time, Thor.

to a parser, in such a way that 5 chunks were parsed with the baseline strategy and the remaining chunks were evenly distributed between the remaining three parsers. This strategy ensures as even a distribution between parsers as possible, while keeping the annotators blind to parser assignments. We avoid the annotators knowing which parser was used, as this could subconciously bias their behaviour.

3 Results

Speed To compare the different parsers as preprocessors for annotation, we need to apply a summary statistic across the times for each annotator, binned by sentence length. We use the median, which is highly resistant to outliers and conceptually simpler than strategies for outlier elimination[4]. Furthermore, to ensure large enough bins, we only consider sentences of length 20 or less.

Figure 1 shows the evolution of annotation time as a function of sentence length for Odin for all four parsers, and Figure 2 the corresponding graphs for Thor. It is clear that, although Odin consistently uses less time to annotate sentences than Thor, the different parsers are ranked identically, and the relative speed-up of the higher quality parsers is similar for both annotators.

Agreement To measure agreement we study the LAS and UAS we get from comparing Odin and Thor's annotations. Artstein and Poesio (2008) argue strongly in favour of using a chance-corrected measure of agreement, but the measures they present are applicable to categorical data, not structured data such as syntactic data. Thus, simple agreement measures are the standard measures in syntax (Hajič, 2004; Miltsakaki et al., 2004; Maamouri et al., 2008). As mentioned in Section 2, only 400 sentences were doubly annotated. Ideally, we would have liked to have all the texts doubly annotated, but external constraints on the annotation effort limited us to the set at hand.

Parser	*n*	UAS	LAS
Baseline	10	99.1%	99.1%
Danish	130	96.3%	94.0%
Swedish	110	96.1%	94.4%
Norwegian	150	96.8%	95.3%

Table 3: Annotator agreement. *n* sentences, unlabelled (UAS) and labelled (LAS) attachment.

Table 3 shows the unlabelled and unlabelled accuracies on the doubly annotated dataset, along with the number of sentences in each dataset. Due to the random distribution of sentences, only a single baseline chunk was in the first 400 sentences, making it hard to draw conclusions on the quality obtained with that strategy. The imbalance is less severe for the other parsers, but the Norwegian set is still almost 50% larger than the Swedish one. The agreement on the baseline set is quite surprising, with only a single token out of 115 receiving different heads and all tokens having the same dependency relation. Unlabelled agreement is lower by about three percentage points on the three remaining datasets, with no real variation in terms

[4]Nor does it assume normality, which would be inappropriate for timing data, unlike most outlier detection methods.

of parser performance, and labelled agreement is somewhat lower again, indicating some level of disagreement over dependency relations.

4 Analysis

Our results are clearest for the question of how time used to annotate is affected by preprocessing quality. The Danish parser halves the time required to annotate sentences compared to the baseline; already an important gain. The Norwegian parser cuts the time in half again, with the Swedish parser between the two. Based on the learning curves in Skjærholt and Øvrelid (2012), a parser with performance equivalent to the Danish parser (70% UAS) can be obtained with about 50 annotated sentences, and the 80% UAS of the Swedish parser is reachable with about 200 sentences.

Given the limited amount of data available for our study of agreement, it is hard to make solid conclusions, but it does appear that head selection is virtually unchanged by parser performance, while there may be some increase in agreement on dependency relation labels, from 96.0% with the Danish parser, to 96.5% and 97.1% with the Swedish and Norwegian parsers. Agreement is extremely high for both heads and labels on the data preprocessed with the baseline parser, but based on 10 sentences, it is impossible to say whether this is a fluke or a reasonable approximation of the value we would get with a larger sample.

The unchanged agreement score suggests that the annotators are not unduly influenced by a better parser. An increase in agreement would not be an unambiguously positive result though; a positive interpretation would be that the annotators' work is closer to the Platonic ideal of a correct analysis of the corpus, but a less charitable interpretation is that the annotators are more biased by the parser. Furthermore, the very high agreement for the baseline parser is potentially worrying if the result remains unchanged by a larger sample. This would indicate that in order to get the best quality annotation, it is necessary to start from a virtually unprocessed corpus, which would require four times as much time as using a 90% UAS parser for preprocessing, based on our data.

5 Conclusions

Given the time-consuming nature of linguistic annotation, higher annotation speed is an obvious good for any annotation project as long as the annotation quality doesn't degrade unacceptably. Based on the results obtained in our study, it is clear that the speed-up to be had from a good dependency parser is important, to the extent that when annotating it is a very bad idea to not use one. Further, based on the learning curves presented in Skjærholt and Øvrelid (2012), it seems that parser adaptation with a view to preprocessing for annotation is primarily useful in the earliest stages of an annotation effort as the learning curves show that once 100 sentences are annotated, a parser trained on that data will already be competitive with a cross-lingual parser for Norwegian. Other languages may require more data, but the amount required is most likely on the same order of magnitude. If same-language data are available, a parser trained on that may last longer.

As regards annotator agreement, our results show that head selection as measured by unlabelled accuracy is unchanged by parser accuracy. Agreement as measured by labelled accuracy increases somewhat with increased parser performance, which indicates that agreement on labels increases with parser performance. The agreement results for our baseline parser are extremely high, but given that we only have ten sentences to compare, it is impossible to say if this is a real difference between the baseline and the other parsers.

5.1 Future work

There are a number of things, particularly relating to annotator agreement we would like to investigate further. Chief of these is the lack of a chance corrected agreement measure for dependency syntax. As mentioned previously, no such measure has been formulated as most agreement measures are most naturally expressed in terms of categorical assignments, which is a bad fit for syntax. However, it should be possible to create an agreement measure suitable for syntax.

We would also like to perform a deeper study of the effects of preprocessing on agreement using a proper measure of agreement. The results for our baseline strategy are based on extremely little data, and thus it is hard to draw any solid conclusions. We would also like to see if different groups of annotators are influenced differently by the parsers. Our annotators were both trained linguists, and it would be interesting to see if using lay annotators or undergraduate linguistics students changes the agreement scores.

References

Ron Artstein and Massimo Poesio. 2008. Inter-Coder Agreement for Computational Linguistics. *Computational Linguistics*, 34(4):555–596.

Fu-Dong Chiou, David Chiang, and Martha Palmer. 2001. Facilitating Treebank Annotation Using a Statistical Parser. In *Proceedings of the first international conference on Human language technology research*, pages 1–4.

Karën Fort and Benoît Sagot. 2010. Influence of Preannotation on POS-tagged Corpus Development. In Nianwen Xue and Massimo Poesio, editors, *Proceedings of the fourth linguistic annotation workshop*, pages 56–63, Stroudsburg. Association for Computational Linguistics.

Jan Hajič. 2004. Complex Corpus Annotation: The Prague Dependency Treebank. Jazykovedný ústav Ľ. Štúra, SAV.

Matthias Trautner Kromann. 2003. The Danish Dependency Treebank and the DTAG Treebank Tool. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories*, pages 217–220. Växjö University Press.

Mohamed Maamouri, Ann Bies, and Seth Kulick. 2008. Enhancing the Arabic Treebank : A Collaborative Effort toward New Annotation Guidelines. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, pages 3192–3196. European Language Resources Association.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2004. The Penn Discourse Treebank. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 2237–2240. European Language Resources Association.

Joakim Nivre, Jens Nilsson, and Johan Hall. 2006. Talbanken05 : A Swedish Treebank with Phrase Structure and Dependency Annotation. In *Proceedings of the fifth international conference on Language Resources and Evaluation*.

Joakim Nivre, Jens Nilsson, Johan Hall, Atanas Chanev, Gülşen Eryiğit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Arne Skjærholt and Lilja Øvrelid. 2012. Impact of treebank characteristics on cross-lingual parser adaptation. In Iris Hendrickx, Sandra Kübler, and Kiril Simov, editors, *Proceedings of the 11th international workshop on treebanks and linguistic theories*, pages 187–198, Lisbon. Edições Colibri.

Anders Søgaard. 2011. Data point selection for cross-language adaptation of dependency parsers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 682–686.

Daniel Zeman and Philip Resnik. 2008. Cross-Language Parser Adaptation between Related Languages. In Anil Kumar Singh, editor, *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 35–42, Hyderabad, India. Asian Federation of Natural Language Processing.