Procesamiento de Lenguaje Natural, Revista nº 48 marzo de 2012, pp 21-27 recibido 15-12-11 revisado 12-03-12 aceptado 18-03-12

Are the existing training corpora unnecessarily large?

¿Son los Corpora de dependencias innecesariamente grandes?

Miguel Ballesteros†, Jesús Herrera†, Virginia Francisco‡, Pablo Gervás‡
†Departamento de Ingeniería del Software e Inteligencia Artificial
‡Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
C/ Profesor José García Santesmases s/n, Madrid, Spain
{miballes, jesus.herrera, virginia}@fdi.ucm.es, pgervas@sip.ucm.es

Resumen: El tamaño de los corpora de entrenamiento ha sido siempre uno de los cuellos de botella de los analizadores de dependencias, tanto en términos de optimización como en términos de precisión. En previos estudios nos dimos cuenta que los corpora pueden contener proporciones significativas de datos redundantes al nivel de árboles sintácticos. Dado que el desarrollo de estos tipos de corpora requiere un gran esfuerzo consideramos que un proceso apropiado para seleccionar las frases que se incluyen en el producto final pueden proporcionar sistemas entrenados con los mismos resultados (o incluso mejores) utilizando menor cantidad de frases. Este argumento se demuestra en el estudio llevado a cabo que se expone en este artículo.
Palabras clave: Análisis sintáctico de dependencias, CoNLL Shared Tasks, Diseño de corpora, Optimización.

Abstract: This paper addresses the problem of optimizing the training treebank data because the size and quality of the data has always been a bottleneck for the purposes of training. In previous studies we realized that current corpora used for training machine learning–based dependency parsers contain a significant proportion of redundant information at the syntactic structure level. Since the development of such training corpora involves a big effort, we argue that an appropriate process for selecting the sentences to be included in them can result in having parsing models as accurate as the ones given when training with bigger – non optimized corpora (or alternatively, bigger accuracy for an equivalent annotation effort). This argument is supported by the results of the study we carried out, which is presented in this paper. Therefore, this paper demonstrates that the training corpora contain more information than needed for training accurate data–driven dependency parsers.
Keywords: Dependency parsing, CoNLL Shared Tasks, Design principles for Treebanks, Optimization.

1 Introduction

Over the last decade research on dependency parsing has focused on English, but in recent years treebanks for other languages have become available and a significant number of languages have been addressed by Machine Learning–based parsers.

The goal of the present work was to test whether the reduction of current corpora may be possible. If so, the immense effort usually needed for developing annotated corpora could be reduced.

A tool for testing our proposals had to be chosen. Since MaltParser (Nivre et al., 2007) is a highly accurate tool in which we are experiencied, it was finally the system we used. For the present work we used the latest version of MaltParser which is dated May 2011.

2 Dependency Parsing: Related Work

There has been a surge of interest in dependency parsing, motivated both by the efficiency and the potential usefulness of bilexical relations in disambiguation. This fact motivated us to develop and simplify the process of developing corpora, training models and parsing data–sets with the emphasis on gathering data which will lead to obtaining stability in parsing performance more quickly.

ISSN 1135-5948

2.1 The CoNLL–X Shared Task

Every year the CoNLL conference features a shared task whose 10th edition was devoted to Multilingual Syntactic Dependency parsing. The aim of this task was to extend the state of the art available at that time in Dependency Parsing. Participants were asked to label dependency structures by means of fully automatic dependency parsers. This Shared Task provided a benchmark for evaluating the participating parsers in 13 languages. Systems were scored with the following token–based measures: LAS, UAS and LA.

For the purposes of the CoNLL–X Shared Task 13 annotated source corpora, one for each language proposed, were provided; we used all of them to develop our experiment: **Arabic** (Hajič et al., 2004), **Czech** (Böhmová et al., 2003), **Danish** (Kromann, 2003), **Slovene** (Džeroski et al., 2006), **Swedish** (Nilsson, Hall, and Nivre, 2005), **Turkish** (Oflazer et al., 2003), **Chinese** (Chen et al., 2003), **Dutch** (van der Beek et al., 2002) , **German** (Brants et al., 2002), **Japanese** (Kawata and Bartels, 2000), **Portuguese** (Afonso et al., 2002), **Bulgarian** (Simov et al., 2005) and **Spanish** (Palomar et al., 2004). In Table 1 we show the sizes of the training corpora.

3 Raising our Hypothesis: Why do we consider training corpora sizes?

Previous works such as (Ballesteros et al., 2010) showed that similar training corpora (i.e., with a similar size in wordforms and a similar distribution of sentences according to their lengths), used to train MaltParser for Spanish, produce models that achieve similar maximum and minimum parsing accuracy values. For this aspect MaltParser shows stable behaviour.

Some authors, such as (Nivre et al., 2007) or (Herrera and Gervás, 2008) have documented signs that the size of the training corpus does not guarantee high parsing accuracy by itself. A large training corpus statistically permits the presence of a wider range of samples, but equally permits the presence of elements that could induce noise when training samples are not selected one by one. These findings suggest that the training corpora of the CoNLL–X Shared Task probably contain redundant information that does not contribute at all to dependency parsing accuracy when training MaltParser. Taking into account that a parser is trained not only on structure and lexical items, but also on their frequency in the training set, we wanted to analyze if we could eliminate some samples from the corpora without affecting training on those issues.

A reduction of the training corpora results in an important reduction in execution time. Considering that MaltParser is an efficient data–driven dependency parser, it is possible to perform parsing in linear time for projective dependency trees and in quadratic time for arbitrary trees (non-projective) (Bosco et al., 2010), which means that a reduction of N% of the nodes present in the training corpus makes a reduction of N% of execution time in the linear case. If we consider the cuadratic case the reduction of N% of the nodes is more remarkable, the execution time is reduced in x% (with $x = ((100 - N)/10)^2$). When we are considering thousand (even millions) of wordforms this is absolutely signficant.

One more important fact when building such kinds of training corpora is their production cost. As an example of the effort necessary for building a dependency annotated corpus, Prokopis Prokopidis et al. reported in (2005) the process related to the Greek Dependency Treebank (GDT), which was used as the training corpus for Greek in the CoNLL Shared Task 2007. The development of this corpus took a full month's work for thirty annotators to reach a final amount of 70,000 words. Regarding our corpora of interest, if we consider the amount of sentences included in them, we can suppose that the work done to annotate all the training corpora of the CoNLL–X Shared Task was hard and really extensive.

In summary, taking into account these previous works, our **hypothesis** is that it is possible to create an equally effective training corpus by removing all those sentences that contain information already present in other samples, when this redundancy is somehow not useful to the trained system. In our work we decided to focus our efforts on **MaltParser**, because the access to MaltParser configurations for each language is easier [1], the execution time (for training and parsing)

[1] http://maltparser.org/userguide.html

	Arabic	Bulgarian	Chinese	Czech	Danish	Dutch	German	Japanese	Portuguese	Slovene	Spanish	Swedish	Turkish
#Sentences	1,479	12,823	57,333	72,703	5,190	13,349	39,216	17,044	9,071	1,534	3,306	11,042	4,997
#Wordforms	54,379	190,217	338,897	1,249,408	94,386	195,069	699,610	151,461	206,678	28,750	89,334	191,467	57,510

Table 1: Number of sentences and wordforms of each training corpus of the CoNLL–X Shared Task

is faster, and finally, it is one of the most established ways of nowadays dependency parsing text.

In the next Sections we show an experiment that demonstrates that the existing training corpora are much too big and we suggest some ideas on how we can reduce their sizes.

4 *Demonstrating our Hypothesis*

To analyze the effect of training corpus size on parsing accuracy we incrementally built a training corpus (for each language) and evaluated parsing performance for each model trained.

4.1 Design of the Experiment

We divided each corpus into 15 smaller pieces, the first 5 using the first 50% of wordforms from each corpus and the second 10 using the second 50% of wordforms from each corpus to show the effect on accuracy when training with incremental training corpora:

This means that for every corpora we divided:

- The first set conforming to the first 50% of wordforms was divided into 5 smaller pieces, each one containing 10% of the wordforms.
- The second set conforming to the second 50% of wordforms was divided into 10 smaller pieces, each one containing 5% of the wordforms.

Each piece respected the average sentence length of the whole corpus. This means that every small piece contained (more or less) the same number of wordforms and the average sentence length was the same for every piece.

Using these 15 small pieces, our experiment was carried out as follows:

- In each iteration we added the next unused piece starting with the first 10% and ending with the last 5%, constructing the whole initial corpus in the last iteration (100%).
- We trained with the resulting incremental corpora and evaluated with the section of the corpora provided as test sets in the CoNLL–X Shared Task.
- We iterated while there were subpieces of the corpora that remain unused.

MaltParser performs labelled parsing, thus, these are the two evaluation measures that we consider in the present work: Labelled Attachment Score[2] (LAS) and Labelled Complete–Match [3] (LCM).

4.2 Results of the Experiment

In this Section we show the results of the experiment described in the previous Section.

4.2.1 Token–Based Results

The results of the experiment for LAS are shown in Table 2, also in Figure 1 we show the behaviour demonstrated by the models for LAS.

After systematic study of the accuracy increment given when incrementing the size of the training corpus while maintaining the distribution of sentence length, we can conclude that over a certain threshold the amount of words in the training corpus does not meaningfully affect the accuracy achieved. In other words, a training corpus containing all kinds of sentence lengths does not significantly contribute to parsing accuracy after a given size.

4.2.2 Complete–Match Results

Although in our experiment we could observe that the lexical level is not as critical as the syntactic level for accurate training due not only to the little number of lexical features considered for training but also to the absence or presence of some phenomena, such as declination in some languages. The words contained in the training corpora are important for inflected languages or morphologically rich languages that use case to

[2]LAS: the percentage of "scoring" tokens for which the system had predicted the correct head and dependency label.

[3]LCM: the percentage of sentences in the test set with correct labelled graph.

Language	10%	20%	30%	40%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
Arabic	1.69	51.08	59.40	61.63	62.11	63.14	**63.01**	63.66	64.66	65.02	66.23	66.60	**66.33**	67.57	**67.42**
Bulgarian	10.81	75.70	82.97	84.90	85.17	85.97	85.89	86.21	86.59	**86.59**	86.94	86.96	87.40	**87.38**	87.57
Chinese	62.09	77.57	81.12	82.82	84.06	84.77	**84.71**	**84.55**	85.17	85.59	85.97	86.63	**86.38**	86.69	86.99
Czech	0.88	71.00	72.98	74.96	**74.96**	75.44	75.70	76.12	**75.8**	76.26	**76.18**	76.42	76.92	**76.92**	77.04
Danish	11.80	77.07	80.18	81.30	81.40	82.33	82.76	**82.66**	83.19	83.59	83.80	83.90	84.29	**84.02**	84.51
Dutch	9.10	66.05	68.40	71.57	72.19	73.35	73.90	**72.97**	**73.79**	**73.63**	**73.43**	74.73	**74.73**	75.01	**74.47**
German	8.74	80.50	82.29	82.74	82.60	83.46	83.81	84.03	84.22	**83.42**	84.70	**84.70**	84.98	85.31	85.33
Japanese	18.43	87.77	89.17	89.80	90.37	90.47	90.63	91.05	91.07	91.21	**91.16**	**91.16**	**90.96**	91.52	91.88
Portuguese	12.29	75.75	77.17	78.99	79.08	79.59	**79.57**	79.91	80.23	**80.13**	80.78	81.12	**81.10**	81.21	**81.21**
Slovene	4.92	49.72	54.20	58.81	59.75	61.36	62.77	63.06	64.04	**63.91**	**64.00**	64.07	**63.98**	64.88	65.56
Spanish	46.36	72.25	75.98	79.30	**78.96**	79.78	79.86	**79.64**	80.22	80.75	**80.36**	**80.75**	81.55	**81.39**	81.87
Swedish	10.21	73.67	75.85	76.62	78.13	78.94	80.14	81.25	82.06	82.10	82.56	83.28	**83.12**	**83.17**	83.50
Turkish	12.45	58.77	61.47	61.98	62.83	63.16	63.25	63.42	63.50	63.68	63.76	64.32	64.58	64.69	64.84
Average	17.48	76.41	80.1	82.12	82.63	83.48	83.83	84.04	84.55	84.66	84.99	85.39	85.53	85.81	86.02

Table 2: General results (LAS) obtained by the iterative models trained with the reduced amount of wordforms corpora. We show in bold the cases in which the result is fewer (or the same) than a previous iteration.

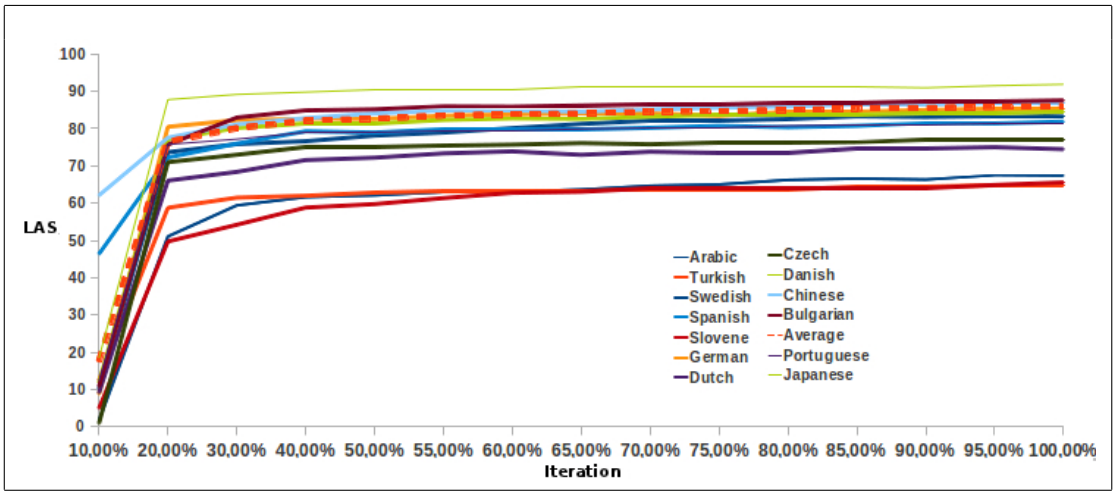

Figure 1: The stable behaviour that the training corpora show when an iterated training experiment is carried out considering LAS.

encode grammatical relations, so wordforms contain information that must be considered, as referred in (Herrera and Gervás, 2008). Given that the errors produced by the models shown in this paper are not the same for every model, complete–match results should be analyzed too. Therefore, the results of the experiment for LCM are shown in Table 3. In Figure 2 we show the behaviour shown by the models for LCM.

As it is shown in the Figure 2, the highest LCM when a significant amount of sentences is already included in the iterative training corpus is achieved very soon, and to include more wordforms in the training corpora does not contribute to final parsing accuracy. When considering the complete–match measures, the results for languages with longer average sentence length are directly affected by this fact, as evidenced in the Figure 2 and Table 3 with the results for Japanese and Chinese.

4.3 Analysis of the Results

As seen in Figures 1 and 2 and Tables 2 and 3, most of the cases show that when a significant amount (normally more than 50%, in some cases less or even more) of the wordforms are included the improvement accuracy obtained when more wordforms are going to be added is not significant. It is important to take into account that 50% of the wordforms of each one of these corpora is a huge amount of information that is somewhat irrelevant for the final accuracy. As it is shown in Table 1 in Section 2, the amount of 50% of the sentences varies from 624,704 in the biggest one (Czech) to 14,375 in the smallest one (Slovene).

Considering the extreme case of the PDT (Czech Prague Dependency Treebank), training a model considering only the first 624,704

Language	10%	20%	30%	40%	50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	100%
Arabic	0.00	1.37	**1.37**	2.05	2.74	**1.37**	**2.74**	**2.05**	**2.05**	**1.37**	**1.37**	9.59	**9.59**	**9.59**	**9.59**
Bulgarian	1.01	13.32	24.12	26.38	26.63	28.90	**28.90**	29.15	30.15	31.16	33.42	**33.17**	33.92	**32.66**	**32.66**
Chinese	29.98	51.56	55.94	59.86	62.97	64.13	64.59	**64.13**	64.70	65.28	66.09	67.13	67.70	67.94	68.51
Czech	0.27	18.08	19.45	20.27	22.46	**22.46**	**22.46**	24.11	24.93	**24.93**	**24.93**	**23.83**	**24.93**	**24.93**	25.02
Danish	0.62	15.84	19.88	22.36	**22.36**	**22.04**	23.60	**22.98**	23.29	24.53	25.16	**24.22**	**24.84**	**23.60**	**24.53**
Dutch	12.18	19.43	19.69	19.95	**19.17**	21.24	21.76	**21.76**	22.02	22.54	23.06	**23.06**	**23.06**	**23.06**	25.38
German	3.36	24.37	29.13	30.25	**29.97**	31.93	33.61	33.89	35.29	**33.61**	**33.61**	**33.61**	**35.01**	**34.35**	**35.29**
Japanese	36.53	68.97	70.52	71.80	**71.80**	72.36	72.92	73.91	**73.91**	**73.62**	74.47	**74.19**	74.90	75.18	75.60
Portuguese	1.74	14.28	**14.28**	15.68	16.37	**16.37**	**16.37**	17.42	**17.42**	**17.42**	18.82	19.16	**19.16**	**19.16**	**19.16**
Slovene	0.00	4.74	**4.74**	6.98	7.98	**7.98**	10.22	10.47	10.72	**10.47**	**10.72**	**9.73**	**9.73**	**10.72**	**10.47**
Spanish	0.00	7.77	12.14	14.56	**14.56**	15.53	**15.53**	**13.59**	**12.62**	14.56	16.50	16.90	18.93	**16.50**	**17.96**
Swedish	6.43	21.85	23.91	25.19	25.44	25.71	28.02	28.53	29.56	30.33	**29.56**	31.62	**31.36**	**31.62**	**31.36**
Turkish	0.16	6.58	7.54	7.70	9.31	9.47	**8.98**	9.79	9.95	10.59	**10.43**	**10.43**	**9.95**	**9.79**	**10.11**
Average	7.69	22.35	25.23	26.92	27.65	28.29	29.14	29.32	29.72	30.03	30.68	31.39	31.92	**31.59**	32.14

Table 3: Labelled Complete Match (LCM) obtained by the iterative models trained with the reduced amount of wordforms corpora. We show in bold the cases in which the result is fewer (or the same) than a previous iteration.

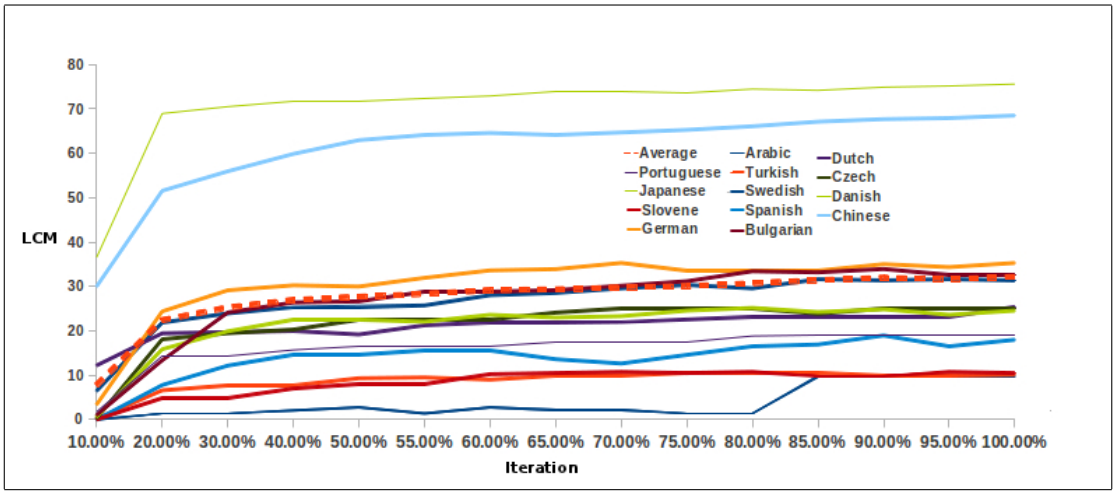

Figure 2: The stable behaviour that the training corpora show when an iterated training experiment is carried out considering Labelled Complete Match (LCM).

wordforms (50%), the performance of the model trained achieves 74.96% LAS and 22.46% LCM, while the whole training corpus model trained with the 100% of the corpus, achieves 77.04% LAS and 25.02% LCM. Thus, an improvement of 74.96 points for LAS is given with the first 624,704 wordforms and the improvement is only 2.08 points for LAS when we added the other 624,704 wordforms. The same thing happens for all languages, taking into account that other languages do not have a corpus as big as the Czech case. Moreover, as is shown in (Buchholz and Marsi, 2006) test corpora consist of about 5,000 wordforms each, and only words (neither punctuation symbols nor other special wordforms) are considered while measuring the score. An increment or a decrement of 1% LAS, only affects 40 tokens. Thus, comparing the results of models trained with 70% of the training corpus and models trained with the whole training corpus, the increment or decrement affects even fewer tokens.

For other languages, when we reach a significant quantity of wordforms, the results are even worse than training with the whole corpus. For Spanish, we observed that using the 90% of the training corpus we obtained 18.93% LCM, while the 95% training model obtained 16.50% LCM, which is more than 2% worse, and using 100% of the training corpus we obtained 17.96% LCM, which is 1% worse. Thus, smaller corpora are even more accurate than bigger ones in some cases. The Spanish case is only an example, all the data shown in bold in the Tables 2 and 3, show the cases in which a model trained with smaller corpora obtained better accuracy (or the same) than the current one for LAS and LCM.

5 Conclusions and Future Work

Due to the results of the experiment, where it is shown that corpora consisting of only 50% of the original size are as accurate (or at least, with a statistically insignificant improvement) as the originals. We can conclude that the study presented in this paper shows that the training corpora used for training data–driven dependency parsers are unnecessarily large. Thus, the results presented should encourage the development of more efficient processes for building training corpora for dependency parsing. This way, the unnecessary effort to label a number of sentences could be avoided by carefully selecting the most convenient ones, according to their syntactic structures. Of course previous dependency tagged corpora are very useful, because they can be used for other, different purposes and, as seen, the results given when used as training corpora for dependency parsing are as good as the ones reached when training with reduced corpora.

So the study shown in this paper should be understood as a justification for the development of new, more efficient processes for building training corpora for dependency parsing because it seems that there are repeated syntactic structures in every training corpora which are useless for data–driven dependency parsers. When speaking of repeated syntactic structures we mean a pair of sentences of the same length that share the whole syntactic structure, or even, a pair of sentences, one of them shorter than the other one, for which the syntactic structure of the shorter one is completely included in the syntactic structure of the larger one.

Therefore, we would suggest for future corpora developments the following ideas:

- Not to repeat structures, but to take into account that for some languages it is also useful not to repeat substructures.
- To include sentences in the widest possible range of lengths. Also, if the corpus is oriented to training a certain system, it will be useful to consider the behaviour of such a system when trained with larger or shorter sentences.

This research is focused on MaltParser and the languages present in the CoNLL–X Shared Task, but similar analyses could be accomplished to study other languages, corpora and/or parsers.

Acknowledgments

This research is funded by the Spanish Ministry of Education and Science (TIN2009-14659-C03-01 Project), Universidad Complutense de Madrid and Banco Santander Central Hispano (GR58/08 Research Group Grant).

References

Abeillé, Anne, editor. 2003. *Treebanks: Building and Using Parsed Corpora*, volume 20 of *Text, Speech and Language Technology*. Kluwer Academic Publishers, Dordrecht.

Afonso, Susana, Eckhard Bick, Renato Haber, and Diana Santos. 2002. Floresta sintá(c)tica: A treebank for Portuguese. In *LREC 2002*.

Ballesteros, Miguel, Jesús Herrera, Virginia Francisco, and Pablo Gervás. 2010. Improving Parsing Accuracy for Spanish using Maltparser. *SEPLN*, 44:83–90, 05/2010.

Böhmová, A., J. Hajič, E. Hajičová, and B. Hladká. 2003. The PDT: a 3-level annotation scenario. In Abeillé (Abeillé, 2003), chapter 7.

Bosco, Cristina, Simonetta Montemagni, Alessandro Mazzei, Vincenzo Lombardo, Felice dell'Orletta, Alessandro Lenci, Leonardo Lesmo, Giuseppe Attardi, Maria Simi, Alberto Lavelli, Johan Hall, Jens Nilsson, and Joakim Nivre. 2010. Comparing the influence of different treebank annotations on dependency parsing. In *LREC*.

Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The tiger treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories (TLT)*.

Buchholz, Sabine and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *CoNLL-X '06: Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 149–164, Morristown, NJ, USA. Association for Computational Linguistics.

Chen, Keh-Jiann, Chi-Ching Luo, Ming-Chung Chang, Feng-Yi Chen, Chao-Jan Chen, Chu-Ren Huang, and Zhao-Ming Gao. 2003. Sinica treebank: Design criteria, representational issues and implementation. In Abeillé (Abeillé, 2003), chapter 13, pages 231–248.

Džeroski, S., T. Erjavec, N. Ledinek, P. Pajas, Z. Žabokrtsky, and A. Žele. 2006. Towards a Slovene dependency treebank. In *In Proc. Int. Conf. on Language Resources and Evaluation (LREC*.

Hajič, Jan, Otakar Smrž, Petr Zemánek, Jan Šnaidauf, and Emanuel Beška. 2004. Prague Arabic dependency treebank: Development in data and tools. pages 110–117.

Herrera, J. and P. Gervás. 2008. Towards a Dependency Parser for Greek Using a Small Training Data Set. *Journal of the Spanish Society for Natural Language Processing (SEPLN)*, 41:29–36.

Kawata, Y. and J. Bartels. 2000. Stylebook for the Japanese treebank in VERBMOBIL. Verbmobil-Report 240, Seminar für Sprachwissenschaft, Universität Tübingen.

Kromann, Matthias T. 2003. The Danish dependency treebank and the underlying linguistic theory. Växjö, Sweden.

Nilsson, Jens, Johan Hall, and Joakim Nivre. 2005. MAMBA meets TIGER: Reconstructing a Swedish treebank from antiquity. In *Proc. of the NODALIDA Special Session on Treebanks*.

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

Oflazer, Kemal, Bilge Say, Dilek Zeynep Hakkani-Tür, and Gökhan Tür. 2003. Building a Turkish treebank. In Abeillé (Abeillé, 2003), chapter 15.

Palomar, M., M. Civit, A. Díaz, L. Moreno, E. Bisbal, M. Aranzabe, A. Ageno, M.A. Martí, and Navarro. 2004. 3lb: Construcción de una base de datos de árboles sintáctico–semánticos para el catalán, euskera y español. In *Proceedings of the XX Conference of the Spanish Society for Natural Language Processing (SEPLN)*, pages 81–88. Sociedad Española para el Procesamiento del Lenguaje Natural.

Prokopidis, P., E. Desypri, M. Koutsombogera, H. Papageorgiou, and S. Piperidis. 2005. Theoretical and Practical Issues in the Construction of a Greek Dependency Treebank. In *Proceedings of The Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005), Barcelona, Spain*, pages 149–160.

Simov, Kiril, Petya Osenova, Alexander Simov, and Milen Kouylekov. 2005. Design and implementation of the Bulgarian HPSG-based treebank. *Journal of Research on Language and Computation – Special Issue*, 2(4):495–522, December.

van der Beek, Leonoor, Gosse Bouma, Robert Malouf, and Gertjan van Noord. 2002. The Alpino dependency treebank. In *Computational Linguistics in the Netherlands (CLIN)*. Rodopi.