Studying the Learning Curves of a Statistical Dependency Parser for Four Languages

Atanas Chanev

Department of Cognitive Sciences
University of Trento
ITC-irst Povo-Trento
via Matteo del Ben 5, 38068 Rovereto, Italia
chanev@form.unitn.it

Abstract

Multilingual dependency parsing is gaining popularity in recent years for several reasons. Dependency structures are more adequate for languages with freer word order than the traditional constituency notion. There is a growing availability of dependency treebanks for new languages. Broad coverage statistical dependency parsers are available and easily portable to new languages. Dependency parsing can provide useful contributions in areas such as information extraction, machine translation and question answering, among others. In addition, syntactic head-dependent pairs are a good interface between the traditional phrase structures and semantic theta roles. In this paper we present the learning curves of a statistical dependency parser for four languages: Arabic, Bulgarian, Italian and Slovene. We discuss issues that mostly concern the employed annotation scheme for each treebank with an emphasis on coordinated structures. We also investigate how these issues are related to the learning curve for each language.

Preučevanje krivulje učenja statističnega odvisnostnega razčlenjevalnika za štiri jezike

Večjezično odvisnostno skladenjsko razčlenjevanje postaja v zadnjih letih vse bolj privlačno zaradi vrste razlogov. Odvisnostne strukture so za jezike s prostejšim besednim redom primernejše kot pa tradicionalne, ki temeljijo na konstituentih, poleg tega pa je na voljo vse več odvisnostnih drevesnic za nove jezike. Statistični odvisnostni razčlenjevalniki s širokim pokritjem so dostopni in lahko prenosljivi na nove jezike. Odvisnostno razčlenjevanje je lahko koristen prispevek področjem, kot so luščenje podatkov, strojno prevajanje in sistemi za odgovarjanje na vprašanja. Poleg tega so skladenjski pari jedro-odvisnica dobri vmesniki med tradicionalno frazno strukturo in pomenskimi vlogami. V članku predstavimo krivulje učenja statističnega odvisnostnega razčlenjevalnika za štiri jezike: arabskega, bolgarskega, italijanskega in slovenskega. Razpravljamo o vprašanjih, ki se dotikajo predvsem uporabe označevalne sheme za vsako drevesnico s poudarkom na zgradbi priredij. Preučimo tudi, kako so ta vprašanja povezana s krivuljo učenja za vsakega od jezikov.

1. Introduction

Contrary to a constituency (or phrase structure) grammar, a dependency grammar (e.g. (Mel'čuk, 1988)) does not view syntactic structures as nested sets of constituents but as a set of binary head-dependent relations. In most dependency grammar formalisms there are several restrictions for the dependency relations: They should build up a connected acyclic graph; For each dependent there should be only one head; There should be a single word in the sentence without a head – the root word. A syntactic label, such as subject, object etc. is usually associated with each relation in the graph.

Projectivity is another issue that is often considered as a constraint to dependency graphs. A simple non-formal definition for projectivity of a connected dependency graph is: if one connects the root word of a sentence with an artificial root placed before the first word, there should not be crossing dependency arcs. While most of the dependency parsers can parse only projective structures, the need for non-projective relations is recognised in nearly all dependency treebank annotation schemes.

State-of-the art statistical dependency parsers have been evaluated on 13 different treebanks (for 13 different languages) at the CoNLL-X shared task on statistical dependency parsing (Buchholz and Marsi, 2006)[1]. While the treebanks had been parsed with many parsers, all the parsers had been an implementation of a limited number of parsing models.

A good multilingual dependency parser should be robust enough so that its accuracy would not decrease when it is ported to another language / another treebank. In practice this is a rarely observed quality, especially when common treebank annotation schemes are based on considerably different linguistic assumptions. The multilingual dependency parsing task is evaluation of the ability of parsers to be ported easily to new languages. But it is also evaluation of the eligibility of treebank annotation schemes to encode linguistic phenomena so that the treebanks can be easily parsed using a statistical dependency parser.

A new direction in designing parsers is using evidence from Psycholinguistics. (Hale, 2001) and (Lesmo et al., 2002), for example, use a psychologically motivated tree pruning and implement incremental processing strategies in their parsers. Incrementality is also addressed in (Nivre, 2004).

This paper gives the learning curves of a statistical dependency parser – the Malt parser (Nivre et al., 2006), for four languages: Arabic, Bulgarian, Italian and Slovene.

[1] http://nextens.uvt.nl/∼conll/

The treebanks for these languages had been annotated by different research groups, using four different annotation schemes. The parser that we use has a high attachment score (accuracy), it is robust and has a number of features that are psychologically plausible.

The paper is structured as follows: Section 2. gives an overview of the results achieved at the CoNLL-X shared task on dependency parsing (Buchholz and Marsi, 2006) and motivation to use the Malt parser in our experiments. Then, in Section 3. we briefly describe the annotation scheme of each treebank that we give learning curves of. We give a short description of the Malt parser and the parsing feature model that we used in our experiments in Section 4. The learning curves are given and discussed in Section 5. We conclude in Section 6.

2. Statistical Dependency Parsing

The parsers from the CoNLL-X shared task usually implemented two parsing models. In one of them the correct dependency graph was searched for as the maximum spanning tree in a full graph with removed arcs that violate the constraints for a dependency graph (e.g. (McDonald et al., 2006)). Parsers of this kind were able to parse non-projective graphs. However, such parsers are not able to assign correct labels to dependency relations during processing. They do it in a following step.

The other approach is an implementation of the shift-reduce parser (Yamada and Matsumoto, 2003), extended as in e.g. (Nivre, 2005). In this model the dependency graph is built in incremental fashion using a stack for storing the words of the sentence and four actions: shift, reduce, left-arc and right-arc. The parser cannot parse non-projective arcs (except e.g. the implementation described in (Attardi, 2006)) but they can be parsed using a technique known as pseudo-projective parsing (Nivre and Nilsson, 2005).

The difference between the accuracy of the two best parsers: (McDonald et al., 2006) that implements the maximum spanning tree approach and (Nivre et al., 2006) that implements the shift-reduce algorithm is not statistically significant (Buchholz and Marsi, 2006). But (Nivre et al., 2006) is more interesting to us because of its psychological plausibility and the fact that any kind of information can be included directly in feature models for learning.

3. Treebanks

We used four treebanks in our experiments: The Prague Arabic Dependency Treebank (PADT) (Hajič et al., 2004), the BulTreeBank (BTB) (Simov et al., 2005), the Turin University Treebank (TUT) (Bosco, 2004) and the Slovene Dependency Treebank (SDT) (Džeroski et al., 2006). Their annotation schemes are different (with PADT and SDT annotation schemes being quite similar). PADT, TUT and SDT are original dependency treebanks while BTB was converted from Head-driven Phrase Structure Grammar (HPSG) format to dependency graphs in (Chanev et al., 2006). We give short descriptions of the treebanks below and summarize their features in Table 1.

Languages:	Ar	Bg	It	Sl
Tokens	59,752	196,151	44,616	35,140
Sentences	1,606	13,221	1,500	1,936
T. per sen.	37.2	14.8	27.7	18.2
PoS set	21	570	90	30
Dep. set	27	20	18	26
Randomized	no	no	yes	no
DG	yes	no	yes	yes

Table 1: Treebank properties. (DG = Dependency Grammar)

3.1. The Prague Arabic Dependency Treebank

We used the CoNLL-X shared task version of the PADT[2] which slightly differs from the original treebank. It is separated in training (1,460 sentences; 54,379 tokens) and test (146 sentences; 5,373 tokens) set. The number of part-of-speech tags and the number of dependency tags are respectively 21 and 27. The average number of tokens per sentence is 37.2. The PADT annotation scheme is closely related to the one of the Prague Dependency Treebank (PDT) (Hajič, 1998).

One of the idiosyncrasies of the PDT annotation scheme is the fact that the root of a sentence is not an ordinary word but an artificial token whose position is e.g. before the first word of the sentence. What should have been the root of a sentence (i.e. the only word that does not have a head) points to the artificial root together with end-of-sentence punctuation. As the artificial root is not included in the CoNLL-X data, one has to learn and parse sentences which have more than one root and their dependency graphs are not connected.

Another idiosyncrasy is the treatment of coordinated structures. In PDT-related annotation schemes the coordinating conjunction (or punctuation) is chosen to be the head of the coordinated words.

3.2. The BulTreeBank

BulTreeBank is an HPSG-based treebank but head-dependent relations between words are not stated explicitly. It has been converted to dependency graph representations in (Chanev et al., 2006). We use the CoNLL-X shared task dependency version of the BTB for our results to be comparable to those from the CoNLL-X shared task.

The BulTreeBank is separated in training (10,911 sentences; 159,395 tokens) and test (2,310 sentences; 36,756 tokens) set. The average number of words per sentence is 14.8. The number of part-of-speech labels is 570[3] and the number of dependency labels is 20.

Contrary to the PADT approach all the graphs in the BulTreeBank have one root per graph. Coordinated structures are annotated differently than those in the PADT. In the BTB encoding the first coordinated word is annotated as the head of the coordinating conjunction (or punctuation) and as the head of the second coordinated word.

[2]PADT is distributed by the Linguistic Data Consortium: http://www.ldc.upenn.edu/

[3]We used the original BTB part-of-speech tags.

3.3. The Turin University Treebank

The TUT was not included in the CoNLL-X shared task mainly because of its limited size – 1,500 sentences (44,616 tokens). The average number of tokens per sentence is 27.7. Although the treebank is small and n-fold cross-validation is usually used in such cases, here we report results on a test set of 150 sentences (4,172 tokens) and a training set of 1,350 sentences (37,444 tokens) in order the TUT experiment not to differ from the experiments on the other treebanks in this study.

We used a version of the TUT with removed traces and reduced tag sets (Chanev, 2005). The reduced tag sets comprised 90 part-of-speech tags and 18 dependency tags. Italian dependency tags are semantically ‘deeper’ than those from the other treebanks in this study. All the graphs in the treebank are connected and have only one root per graph. Coordination is annotated with the coordinating conjunction (or punctuation) being head of the second coordinated word and dependent on the first coordinated word. A single coordination dependency tag is used for both of the dependency arcs[4].

3.4. The Slovene Dependency Treebank

SDT has an annotation scheme which is similar to those of the PDT and PADT. We used the CoNLL-X version of the treebank for our results to be comparable with those from the shared task. The data is divided in a training set (1,534 sentences, 28,750 words) and a test set (402 sentences, 6,390 words). The average number of tokens per sentence is 18.2. The number of the part-of-speech tags used in the annotation of SDT is 30. The number of dependency labels is 26. Like in PADT, sentences can have more than one root and coordinated structures are treated with the coordinating conjunction (or punctuation) as the head of the coordinated words.

4. The Parser

We used version 0.4 of the Malt parser[5]. It is related to the shift-reduce dependency parser described in (Yamada and Matsumoto, 2003). There are two different parsing algorithms: arc-eager and arc-standard. In all our experiments we used the arc-eager parsing algorithm because it is more accurate than the arc-standard algorithm[6].

Malt parser does not use an explicit probabilistic grammar but implements a data-driven parsing approach. What is learned is the actions that the shift-reduce parser must take in order to build the dependency graph of the sentence. Two learners are available for that task: Memory-Based Learning (MBL) (Daelemans and den Bosch, 2005) and Support Vector Machines (SVM) (Chang and Lin, 2005). PoS tags, words as well as dependency labels which have already been assigned by the parser on the run can be used in feature models for learning.

In all the experiments we used the SVM learner. We also employed a common feature model – m7 that has proven to outperform the m3 and m4 models. It consists of six part-of-speech features, four dependency features and four lexical features. More information about the parser and feature models can be found in (Nivre, 2005) as well as on the Malt parser web page. The Malt parser team reported the second best result at the CoNLL-X shared task (Nivre et al., 2006) (the difference from the best result is not statistically significant).

[4]This approach differs from the one implemented in the original TUT.

[5]http://w3.msi.vxu.se/∼nivre/research/MaltParser.html

[6]This issue was discussed with Joakim Nivre in personal communication.

5. Results

In this section we list related work, describe preliminary settings, present and discuss the learning curves for Arabic, Bulgarian, Italian and Slovene.

5.1. Previous Studies

Even though constituency parsing is undoubtedly related to dependency parsing, in this section we give only dependency parsing results because they are immediately relevant to the study.

5.1.1. Arabic

The PADT has been learned and parsed by various teams at the CoNLL-X shared task on dependency parsing. Results vary from 50.7% to 66.9% labelled accuracy (Buchholz and Marsi, 2006).

5.1.2. Bulgarian

A dependency version of the BulTreeBank has also been used at the CoNLL-X shared task. Labelled accuracy is within the range 67.6% – 87.6%. Labelled accuracy of 79.5% was reported for another conversion of the original HPSG-based BulTreeBank but those results did not differ significantly from the results reported on the CoNLL-X conversion using the same parser and feature model (79.2%) (Chanev et al., 2006).

5.1.3. Italian

There are not many studies on statistical dependency parsing of Italian mainly because there are not large enough resources to train a parser. We will compare the learning curve for Italian with (Chanev, 2005) where the Malt parser was used together with the MBL learner. The reported accuracy is 81.8%. (Lesmo et al., 2002) describe a rule-based dependency parser for Italian. Even though its evaluation is only partial, accuracy is comparable to (Chanev, 2005).

5.1.4. Slovene

Slovene, like Arabic and Bulgarian, was one of the languages for the CoNLL-X shared task. Results for Slovene varied from 50.7% to 73.4% labelled accuracy (Buchholz and Marsi, 2006). Parsing Slovene was also mentioned in (Chanev, 2005) where the Malt parser with the MBL learner was trained on an old and very small version of the SDT. Labelled accuracy of 58.3% was reported.

5.2. Settings

All the experiments were performed on training / test sets with gold standard PoS tags. The same feature model and the same learning and parsing settings were used in all the tests with the exception of an option that allowed many roots in a sentence that was used only for the Arabic and

Slovene treebanks. The measure that we use is labelled attachment score (labelled accuracy) measured excluding punctuation. We chose this measure for comparison reasons, since it is the measure used in the evaluation of the parsers at the CoNLL-X shared task. However, we also report unlabelled attachment score (unlabelled accuracy) for the biggest data sets for all the treebanks. For a definition of these measures, the reader is referred to (Lin, 1998).

The BulTreeBank learning curve is set for training sets that start from 1,000 sentences and increase up to the full size of the treebank, where at each step the size of the training set is increased by 1,000 sentences. The learning curves for the other languages start from a training set of 600 sentences and the sizes continue to grow up to the full number of sentences of the treebanks with increase of 200 sentences at each step. The sentences from the Italian treebank were randomized. However, the other treebanks were not, due to CoNLL-X shared task compatibility reasons.

Two additional learning curves are included for Arabic and Slovene after a simple graph transformation on the coordinated structures was applied on the training sets for these languages. Parsing output was then converted back to the original coordination encoding and evaluated on the gold standard PADT and SDT. These learning curves are shown with squares on the graphics for Arabic and Slovene on Figure 1.

A description of the coordination transformation procedure follows:

Coordinated structures are identified by the dependency label of the coordinating conjunction (or punctuation) which, according to the PDT annotation scheme, is the head of the coordinated words. If there are two words with the same dependency labels among the dependents, one of them being before the head and the other – after the head, then they are recognised as coordinated. Then the first coordinated word takes the head word of the coordinating conjunction (punctuation) and the coordinating conjunction or punctuation is made to point to the first coordinated word.

The inverted transformation is performed in a similar way. After the coordinated structure is identified, the head of the first coordinated word is transferred to be the head of the coordinating conjunction (or punctuation) and the first coordinated word is made dependent on the coordinating conjunction (or punctuation). Note that the back transformation can be accurate only for properly parsed coordinated structures. It is important that coordinated words have correct labels, otherwise a coordinated structure cannot be easily identified for inverted transformation.

5.3. Learning Curves

The learning curves are given in Figure 1. X-axis in the graphics shows the number of sentences used for training. The measure on the y-axis of the graphics is labelled accuracy. Results reported on data sets with transformed coordination structures for the Arabic and Slovene treebanks are given with squares. The best results are achieved for the biggest training data as shown in Table 2.

For training data of 1,000 sentences labelled accuracies for Bulgarian, Slovene and Arabic are similar. Labelled accuracy for Italian is the best for this size of training data.

Languages:	Ar	Bg	It	Sl
AS_L	*67.4%	81.8%	83.7%	*68.2%
AS_U	*78.0%	86.8%	88.6%	*77.6%

Table 2: Best results for Arabic, Bulgarian, Italian and Slovene. AS_L = labelled attachment score; AS_U = unlabelled attachment score; * = Coordination transformation applied.

If the comparison is done using the unlabelled accuracy measure, the per cent for Bulgarian is lower than those for Arabic and Slovene due to the bigger difference between labelled and unlabelled accuracy for PADT and SDT. Unlabelled accuracy is on the average 11% higher than labelled accuracy for Arabic, 5% for Bulgarian and Italian and 10% for Slovene.

There are a number of reasons for differences in accuracy for the different treebanks, from numbers of tokens per sentence for each treebank to sizes of the tag sets and idiosyncrasies of the annotation schemes. For example, the small number of part-of-speech tags for the Arabic and Slovene treebanks might have been the reason for the lower accuracy, in comparison with the bigger number of PoS tags for the Italian treebank, given that the number of dependency tags is similar in all the three treebanks. In fact, this is not the case. We did an additional experiment on the Italian data. We used a PoS set of only 17 coarse grained tags and labelled accuracy was still above 81%.

The Arabic and Slovene data sets had their transformed versions learned and parsed better than the original ones. The difference is over 1% for nearly all the sets. The biggest training set gives worse results than the second biggest for the transformed Slovene staining data. This is due to loss of accuracy in the inverted transformation.

There are two other factors which are relevant to parsing accuracy. The number of non-projective sentences in each treebank is the first factor. Non-projective arcs cannot be parsed correctly using the Malt parser without employing the pseudo-projective technique described in (Nivre and Nilsson, 2005). The other factor, which is relevant for the Slovene and Arabic treebanks, is the number of coordinated structures.

The number of non-projective trees for the Arabic, Bulgarian, Italian and Slovene treebanks are respectively 175 (10.9%), 962 (7.3%), 91 (6.1%) and 1,289 (66.6%). The number of sentences with coordinated structures in the PADT and SDT are respectively 1,041 (64.8%) and 989 (51.1%).

The overall results for Arabic and Slovene are the worst, compared to the results for the other treebanks. Labelled accuracy for PADT is around 1% smaller than labelled accuracy for the same size data sets for Slovene. It seems that the incremental dependency parser has difficulties with coordination treatment in these annotation schemes, at least for small data sets[7].

[7]The PDT was also one of the hard-to-parse treebanks at the CoNLL-X shared task despite its larger size (Buchholz and Marsi, 2006).

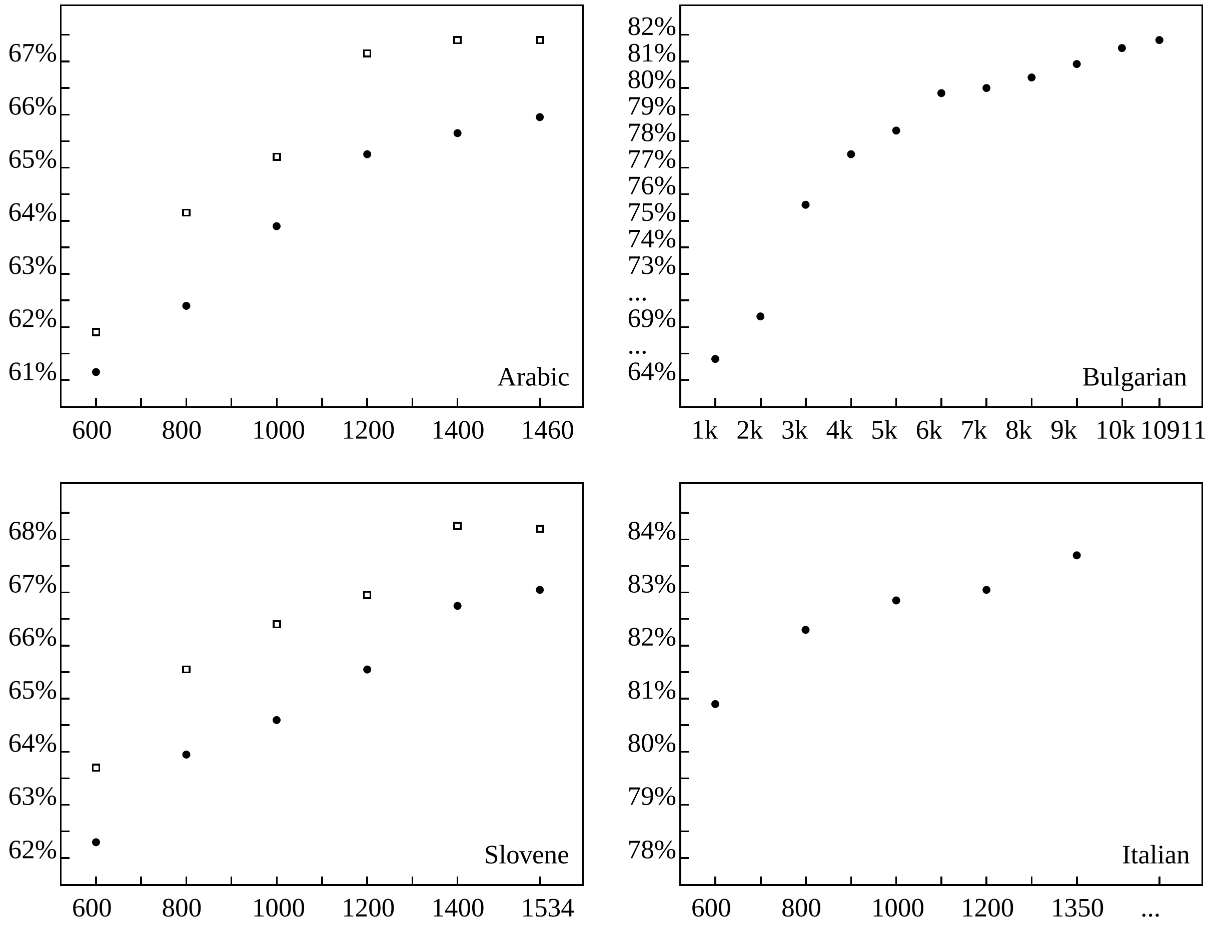

Figure 1: Clockwise: Learning curves for Arabic (top left), Bulgarian, Italian and Slovene (labelled attachment score).

Figure 1. shows that the coordination transformations increased parsing accuracy for the Arabic (and Slovene) data sets. Due to the imperfect back transformation procedure some accuracy has been lost. The number of non-projective sentences in PADT is comparatively small – only 175. The number of sentences with coordination is 1,041. The results for Arabic reported in this paper are slightly higher (0.5%) than the best results reported at the CoNLL-X shared task even though a more sophisticated feature model for the Malt parser was used there.

Results for Bulgarian are lower, compared to the results obtained at the CoNLL-X shared task where the Malt parser had a better feature model and the data was parsed pseudo-projectively. The accuracy that we report is higher than the one reported in (Chanev et al., 2006) because they used an option of the SVM learner which splits the data on smaller parts for faster learning with the cost of decrease in performance.

Compared to the other treebanks the parser learned TUT very well with a limited amount of training data. The reason for the good performance cannot be the number of tokens per sentence (SDT has less and PADT has more tokens per sentence). Sizes of the tag sets are not suspiciously small to be the main reason for the good results on little training data. It may be concluded that the reason for the high accuracies is the treebank annotation scheme. It is different from those of the other treebanks in its 'deeper' syntactic dependency relations. The distance between the dependents and their heads is usually short which facilitates processing.

Compared to (Chanev, 2005) there is an increase of accuracy due to the use of a more advanced feature model for the parser and the better SVM learner. The number of sentences in TUT which have non-projective graphs is very small[8] – only 91. That may have contributed to the high parsing accuracy.

Our results for Slovene somehow lag behind the results for that language which were obtained using the Malt parser at the CoNLL-X shared task. The reasons are the use of a simple feature model for the parser and the big number of non-projective trees in the Slovene treebank (1,289) which we did not parse pseudo-projectively.

Results are on the average 1% higher than those for the PADT. Possibly this difference can be explained with the very small number of tokens per sentence for the SDT – only 18.2, compared to 37.2 for the Arabic treebank. The number of coordinated structures is 989. As in the case

[8]Originally TUT does not have non-projective sentences but after traces were removed in (Chanev, 2005) non-projective arcs were introduced.

with PADT, coordination transformations increased parsing results.

6. Conclusion and Future Work

We presented the learning curves for four different treebanks using the same feature model for learning an incremental statistical dependency parser. We showed that often parsing results differ significantly for different languages and the reasons can be various properties of the concrete treebank. We performed treebank transformations for Arabic and Slovene to report parsing accuracy for Arabic that is slightly higher than the best results reported at the CoNLL-X shared task. We compared the annotation schemes of the treebanks by measuring the extent to which they can be learned and parsed using an incremental parser.

Future work includes investigation of various treebanks to find out which annotation scheme keeps parsing accuracy high for a vast majority of languages. In addition we believe that adding different kind of information to feature models for parsers with incremental architectures can lead to successful broad coverage models of the human sentence parsing mechanism whose implementations must be good multilingual NLP parsers.

Acknowledgements

I thank the organizers of the CoNLL-X shared task for providing the PADT in CoNLL format (through the LDC) as well as for their devoted work on organizing the task. I thank Kiril Simov and Petya Osenova for making the BulTreeBank available, Cristina Bosco – for the TUT and Tomaž Erjavec – for the SDT. I also thank Alberto Lavelli for being my co-advisor, one anonymous reviewer and the Malt parser team at the CoNLL-X shared task (Joakim Nivre, Johan Hall, Jens Nilsson, Gülşen Eryiğit and Svetoslav Marinov).

7. References

G. Attardi. 2006. Experiments with a multilingual non-projective dependency parser. In: *Proc. of the Tenth Conference on Computational Natural Language Learning (CoNLL)*, New York.

C. Bosco. 2004. *A grammatical relation system for treebank annotation*. PhD thesis, University of Turin.

S. Buchholz and E. Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In: *Proc. of the Tenth Conference on Computational Natural Language Learning (CoNLL)*, New York.

A. Chanev, K. Simov, P. Osenova, and S. Marinov. 2006. Dependency conversion and parsing of the BulTreeBank. In: *Proc. of the LREC workshop Merging and Layering Linguistic Information*, Genoa.

A. Chanev. 2005. Portability of dependency parsing algorithms - an application for Italian. In: *Proc. of the workshop Treebanks and Linguistic Theories*, Barcelona.

C.-C. Chang and C.-J. Lin. 2005. LIBSVM: A library for Support Vector Machines. *URL: http://www.csie.ntu.edu.tw/∼cjlin/papers/libsvm.pdf*.

W. Daelemans and A. Van den Bosch. 2005. *Memory-Based Language Processing*. Cambridge University Press.

S. Džeroski, T. Erjavec, N. Ledinek, P. Pajas, Z. Žabokrtsky, and A. Žele. 2006. Towards a Slovene dependency treebank. In: *Proc. of the Fifth International Conference on Language Resources and Evaluation*, Genoa.

J. Hajič, O. Smrž, P. Zemánek, J. Šnaidauf, and E. Beška. 2004. Prague arabic dependency treebank: Development in data and tools. In: *Proc. of the NEMLAR International Conference on Arabic Language Resources and Tools*.

J. Hajič. 1998. Building a syntactically annotated corpus: The Prague Dependency Treebank. In: *Issues of Valency and Meaning*, Prague. Karolinum.

J. Hale. 2001. A probabilistic Earley parser as a psycholinguistic model. In: *Proc. of Second meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh.

L. Lesmo, V. Lombardo, and C. Bosco. 2002. Treebank development: the TUT approach. In: R. Sangal and S.M. Bendre, ed., *Recent Advances in Natural Language Processing*, New Delhi. Vikas Publ. House.

D. Lin. 1998. A dependency-based method for evaluating broad-coverage parsers. *Natural Language Engineering*, 4 (2):97–114.

R. McDonald, K. Lerman, and F. Pereira. 2006. Multilingual dependency analysis with a two-stage discriminative parser. In: *Proc. of the Tenth Conference on Computational Natural Language Learning (CoNLL)*, New York.

I. Mel'čuk. 1988. *Dependency syntax: Theory and practice*. State University of New York Press.

J. Nivre and J. Nilsson. 2005. Pseudo-projective dependency parsing. In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

J. Nivre, J. Hall, J. Nilsson, G. Eryiğit, and S. Marinov. 2006. Labeled pseudo-projective dependency parsing with Support Vector Machines. In: *Proc. of the Tenth Conference on Computational Natural Language Learning (CoNLL)*, New York.

J. Nivre. 2004. Incrementality in deterministic dependency parsing. In: *Incremental Parsing: Bringing Engineering and Cognition Together, Workshop at ACL-2004*, Barcelona.

J. Nivre. 2005. *Inductive Dependency Parsing of Natural Language Text*. PhD thesis, University of Växjö.

K. Simov, P. Osenova, A. Simov, and M. Kouylekov. 2005. Design and implementation of the Bulgarian HPSG-based treebank. *Journal of Research on Language and Computation – Special Issue*, str. 495–522.

H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with Support Vector Machines. In: *Proc. of IWPT*, Nancy.