Cross-Language Parser Adaptation between Related Languages

Daniel Zeman
Univerzita Karlova
Ústav formální a aplikované lingvistiky
Malostranské náměstí 25
CZ-11800 Praha
zeman@ufal.mff.cuni.cz

Philip Resnik
University of Maryland
Department of Linguistics and
Institute for Advanced Computer Studies
College Park, MD 20742, USA
resnik@umd.edu

Abstract

The present paper describes an approach to adapting a parser to a new language. Presumably the target language is much poorer in linguistic resources than the source language. The technique has been tested on two European languages due to test data availability; however, it is easily applicable to any pair of sufficiently related languages, including some of the Indic language group. Our adaptation technique using existing annotations in the source language achieves performance equivalent to that obtained by training on 1546 trees in the target language.

1 Introduction

Natural language parsing is one of the key areas of natural language processing, and its output is used in numerous end-user applications, e.g. machine translation or question answering. Unfortunately, it is not easy to build a parser for a resource-poor language. Either a reasonably-sized syntactically annotated corpus (treebank) or a human-designed formal grammar is typically needed. These types of resources are costly to build, both in terms of time and of the expenses on qualified manpower. Both also require, in addition to the actual annotation process, a substantial effort on treebank/grammar design, format specifications, tailoring of annotation guidelines etc; the latter costs are rather constant no matter how small the resulting corpus is.

In this context, there is the intriguing question whether we can actually build a parser without a treebank (or a broad-coverage formal grammar) *of the particular language*. There is some related work that addresses the issue by a variety of means. Klein and Manning (2004) use a hybrid unsupervised approach, which combines a constituency and a dependency model, and achieve an unlabeled F-score of 77.6% on Penn Treebank Wall Street Journal data (English), 63.9% on Negra Corpus (German), and 46.7% on the Penn Chinese Treebank. Note that in all these experiments they restrict themselves to sentences of 10 words or less. Bod (2006) uses unsupervised data-oriented parsing; the input of his parser contains manually assigned gold-standard tags. He reports 64.2% unlabeled F-score on WSJ sentences up to 40 words long.[1]

Hwa et al. (2004) explore a different approach to attacking a new language. They train Collins's (1997) Model 2 parser on the Penn Treebank WSJ data and use it to parse the English side of a parallel corpus. The resulting parses are converted to dependencies, the dependencies are projected to a second language using automatically obtained word alignments as a bridge, and the resulting dependency trees cleaned up using a limited set of language-specific post-projection transformation rules. Finally a dependency parser for the target language is trained on this projected dependency treebank, and the accuracy of the parser is measured against a gold standard. Hwa et al. report dependency accuracy of 72.1% for Spanish, comparable to a rule-based commercial parser; accuracy on Chinese is 53.9%, the equivalent of a parser trained on roughly 2000 sentences of the Penn Chinese Treebank (sentences ≤40 words, average length 20.6).

[1] On sentences of ≤10 words, Bod achieves 78.5% for English (WSJ), 65.4% for German (Negra) and 46.7% for Chinese (CTB).

Our own approach is motivated by McClosky et al.'s (2006) reranking-and-self-training algorithm, used successfully in adapting a parser to a new domain. One can easily imagine viewing two dialects of a language or even two related languages as two domains of one "super-language". While the vocabulary will certainly differ (due to independently designed orthographies for the two languages) many morphological and syntactic properties may be shared. We trained Charniak and Johnson's (2005) reranking parser on one language and applied it to another closely related language. In addition, we investigated the utility of large but unlabeled data in the target language, and of a large parallel corpus of the two languages.[2]

2 Corpora and Other Resources

The selection of our source and target languages was driven by the need for two closely related languages with associated treebanks. (In a real-world application we would not assume the existence of a target-language treebank, but one is needed here for evaluation.) Danish served as the source language and Swedish as target, since these languages are closely related and there are freely available treebanks for both.[3]

The Danish Dependency Treebank (Kromann et al. 2004) contains 5,507 sentences (average length 18 tokens). The texts come from the Danish Parole Corpus (1998–2002, mixed domain). We used 4,895 sentences for training, 290 for development and 322 for testing (306 not exceeding 40 words).

The Swedish treebank Talbanken05 (Nivre et al. 2006) contains 11,411 sentences (average length 17 tokens). It was converted at Växjö from the much older Talbanken76 treebank, created at the Lund University. Again, the texts belong to mixed domains. We split the data to 10,681 training, 341 development and 389 test sentences, out of which 386 do not exceed 40 words.

Both treebanks are dependency treebanks, while the Charniak-Johnson reranking parser works with phrase structures. For our experiments, we converted the treebanks from dependencies to phrases, using the "flattest-possible" algorithm (Collins et al. 1999; algorithm 2 of Xia and Palmer 2001). The morphological annotation of the treebanks helped us to label the non-terminals. Although the Charniak's parser can be taught a new inventory of labels, we found it easier to map head morpho-tags directly to Penn-Treebank-style non-terminals. Hence the parser can think it's processing Penn Treebank data. The morphological annotation of the treebanks is further discussed in Section 4.

We also experimented with a large body of unannotated Swedish texts. Such data could theoretically be acquired by crawling the Web; here, however, we used the freely available JRC-Acquis corpus of EU legislation (Steinberger et al. 2006).[4] The Acquis corpus is segmented at the paragraph level. We ran a simple procedure to split the paragraphs into sentences and pruned sentences with suspicious length, contents (sequence of dashes, for instance) or both. We ended up with 430,808 Swedish sentences and 6,154,663 tokens.

Since the Acquis texts are available in 21 languages, we can also exploit the Danish Acquis and its alignment with the Swedish one. We use it to study the similarity of the two languages, and for the "gloss" experiment in Section 5.1. Paragraph-level alignment is provided as part of Acquis and contains 283,509 aligned segments. Word-level alignment, needed for our experiment, was obtained using GIZA++ (Och and Ney 2000).

The treebanks are manually tagged with parts of speech and morphological information. For some of our experiments, we needed to automatically retag the target (Swedish) treebank, and to tag the Swedish Acquis. For that purpose we used the Swedish tagger of Jan Hajič, a variant of Hajič's Czech tagger (Hajič 2004) retrained on Swedish data.

3 Treebank Normalization

The two treebanks were developed by different teams, using different annotation styles and guidelines. They would be systematically different even

[2] There are other approaches to domain adaptation as well. For instance, Steedman et al. (2003) address domain adaptation using a weakly supervised method called co-training. Two parsers, each applying a different strategy, mutually prepare new training examples for each other. We have not tested co-training for cross-language adaptation.

[3] We used the CoNLL 2006 versions of these treebanks.

[4] Legislative texts are a specialized domain that cannot be expected to match the domain of our treebanks, however vaguely defined it is. But presumably the domain matching would be even less trustworthy if we acquired the unlabeled data from the web.

if their texts were in the same language, but it is the impact of the language difference, not annotation style differences, that we want to measure; therefore we normalize the treebanks so that they are as similar as possible.

While this may sound suspicious at first glance ("wow, are they refining their test data?!"), it is important to understand why it does not unacceptably bias the results. If our method were applied to a new language, where no treebank exists, trees conforming to the annotation scenario of a treebank of related language would be perfectly satisfying. In addition, note that we apply only systematic changes, mostly reversible. Moreover, the transformations can be done on the training data side, instead of test data.

Following are examples of the style differences that underwent normalization:

DET-ADJ-NOUN. Da: *de norske piger.* Sv:[5] *en gammal institution* ("an old institution") In DDT, the determiner governs the adjective and the noun. The approach of Talbanken (and of a number of other dependency treebanks) is that both determiner and adjective depend on the noun.

NUM-NOUN. Da: *100 procent* ("100 percent") Sv: *två eventuellt tre år* ("two, possibly three years") In DDT, the number governs the noun. In Talbanken, the number depends on the noun.

GENITIVE-NOMINATIVE. Da: *Ruslands vej* ("Russia's way") Sv: *års inkomster* ("year's income"). In DDT, the nominative noun (the owned) governs the noun in genitive (the owner). Talbanken goes the opposite way.

COORDINATION. Da: *Færøerne og Grønland* ("Faroe Islands and Greenland") Sv: *socialgrupper, nationer och raser* ("social groups, nations and races") In DDT, the last coordination member depends on the conjunction, the conjunction and everything else (punctuation, inner members) depend on the first member, which is the head of the coordination. In Talbanken, every member depends on the previous member, commas and conjunctions depend on the member following them.

4 Mapping Tag Sets

The nodes (words) of the Danish Dependency Treebank are tagged with the Parole morphological tags. Talbanken is tagged using the much coarser Mamba tag set (part of speech, no morphology). The tag inventory of Hajič's tagger is quite similar to the Danish Parole tags, but not identical. We need to be able to map tags from one set to the other. In addition, we also convert pre-terminal tags to the Penn Treebank tag set when converting dependencies to constituents.

Mapping tag sets to each other is obviously an information-lossy process, unless both tag sets cover identical feature-value spaces. Apart from that, there are numerous considerations that make any such conversion difficult, especially when the target tags have been designed for a different language.

We take an Interlingua-like (or Inter-tag-set) approach. Every tag set has a *driver* that implements decoding of the tags into a nearly universal feature space that we have defined, and encoding of the feature values by the tags. The encoding is (or aims at being) independent of where the feature values come from, and the decoding does not make any assumptions about the subsequent encoding. Hence the effort put in implementing the drivers is reusable for other tagset pairs.

The key function, responsible for the universality of the method, is `encode()`. Consider the following example. There are two features set, POS = "noun" and GENDER = "masc". The target set is not capable of encoding masculine nouns. However, it allows for "noun" + "com" | "neut", or "pronoun" + "masc" | "fem" | "com" | "neut". An internal rule of `encode()` indicates that the POS feature has higher priority than the GENDER feature. Therefore the algorithm will narrow the tag selection to noun tags. Then the gender will be forced to common (i.e. "com").

Even the precise feature mapping does not guarantee that the *distribution* of the tags in two corpora will be reasonably close. All converted source tags will now fit in the target tag set. However, some tags of the target tag set may not be used, although they are quite frequent in the corpus where the target tags are native. Some examples:

- Unlike in Talbanken, there are no **determiners** in DDT. That does not mean there

[5] These are separate examples from the two treebanks. They are *not* translations of each other!

Bestemmelserne i denne aftale kan ændres og revideres **helt eller delvis efter** fælles
Bestämmelserna i detta avtal får ändras eller revideras **helt eller delvis efter** gemensam
overenskomst mellem parterne.
överenskommelse mellan parterna.

Figure 1. Comparison of matching Danish (upper) and Swedish (lower) sentences from Acquis. Despite the one-to-one word mapping, only the 5 bold words have identical spelling.

are no determiners in Danish – but DDT tags them as pronouns.

- Swedish tags encode a special feature of **personal pronouns,** “subject” vs. “object” form (the distinction between English *he* and *him*). DDT calls the same paradigm “nominative” vs. “unmarked” case.
- Most noun phrases in both languages distinguish just the **common and neuter genders.** However, some pronouns could be classified as masculine or feminine. Swedish tags use the masculine gender, Danish do not.
- DDT does not use special part of speech for **numbers** — they are tagged as adjectives.

All of the above discrepancies are caused by differing designs, not by differences in language. The only linguistically grounded difference we were able to identify is the **supine** verb form in Swedish, missing from Danish.

When not just the tag *inventories,* but also the tag *distributions* have to be made compatible (which is the case of our delexicalization experiments later in this paper), we can create a new *hybrid* tag set, omitting any information specific for one or the other side. Tags of both languages can then be converted to this new set, using the universal approach described above.

5 Using Related Languages

The Figure 1 gives an example of matching Danish and Swedish sentences. This is a real example from the Acquis corpus. Even a non-speaker of these languages can detect the evident correspondence of at least 13 words, out of the total of 16 (ignoring final punctuation). However, due to different spelling rules, only 5 word pairs are stringwise identical. From a parser’s perspective, the rest is unknown words, as it cannot be matched against the vocabulary learned from training data.

We explore two techniques of making unknown words known. We call them *glosses* and *delexicalization*, respectively.

5.1 Glosses

This approach needs a Danish-Swedish (da-sv) bitext. As shown by Resnik and Smith (2003), parallel texts can be acquired from the Web, which makes this type of resource more easily available than a treebank. We benefited from the Acquis da-sv alignments.

Similarly to phrase-based translation systems, we used GIZA++ (Och and Ney 2000) to obtain one-to-many word alignments in both directions, then combined them into a single set of refined alignments using the “final-and” method of Koehn et al. (2003). The refined alignments provided us with two-way tables of a source word and all its possible translations, with weights. Using these tables, we glossed each Swedish word by its Danish, using the translation with the highest weight.

The glosses are used to replace Swedish words in test data by Danish, making it more likely that the parser knows them. After a parse has been obtained, the trees are “restuffed” with the original Swedish words, and evaluated.

5.2 Delexicalization

A second approach relies on the hypothesis that the interaction between morphology and syntax in the two languages will be very similar. The basic idea is as follows: Replace Danish words in training data with their morphological (POS) tags. Similarly, replace the Swedish words in test data with tags. This replacement is called delexicalization. Note that there are now two levels of tags in the trees: the Danish/Swedish tags in terminal nodes, and the Penn-style tags as pre-terminals. The terminal tags are more descriptive because both Nordic languages have a slightly richer morphology than English, and the conversion to the Penn tag set loses information.

The crucial point is that both Danish and Swedish use the same tag set, which helps to deal with the discrepancy between the training and the test terminals.

Otherwise, the algorithm is similar to that of glosses: train the parser on delexicalized Danish, run it over delexicalized Swedish, restuff the resulting trees with the original Swedish words ("relexicalize") and evaluate them.

6 Experiments: Part One

We ran most experiments twice: once with Charniak's parser alone ("C") and once with the reranking parser of Charniak and Johnson, which we label simply Brown parser ("B").

We use the standard `evalb` program by Sekine and Collins to evaluate the parse trees. Keeping with tradition, we report the F-score of the *labeled* precision and recall on the sentences of up to 40 words.[6]

Language	Parser	P	R	F
da	C	77.84	78.48	78.16
	B	78.28	78.20	78.24
da-hybrid	C	79.50	79.73	79.62
	B	**80.60**	**79.80**	**80.20**
sv	C	77.61	78.00	77.81
	B	79.16	78.33	78.74
sv-mamba	C	77.54	78.93	78.23
	B	**79.67**	**79.26**	**79.46**
sv-hybrid	C	76.10	76.04	76.07
	B	78.12	75.93	77.01

Table 1. Monolingual parsing accuracy.

To put the experiments in the right context, we first ran two monolingual tracks and evaluated Danish-trained parsers on Danish, and Swedish-trained parsers on Swedish test data. Both treebanks have also been parsed after delexicalization into various tag sets: Danish gold standard converted to the hybrid sv/da tag set, Swedish Mamba gold standard, and Swedish automatically tagged with hybrid tags.

The reranker helps only slightly, though consistently for all monolingual experiments. Another observation is that delexicalized reranking parsers outperformed lexicalized parsers for both languages. This holds for delexicalization using the gold standard tags (even though the Mamba tag set encodes much less information than the hybrid tags). Automatically assigned tags perform significantly worse.

Our baseline condition is simply to train the parsers on Danish treebank and run them over Swedish test data. Then we evaluate the two algorithms described in the previous section: glosses and delexicalization (hybrid tags).

Approach	Parser	P	R	F
baseline	C	44.59	42.04	43.28
	B	42.94	40.80	41.84
glosses	C	61.85	65.03	63.40
	B	60.22	62.85	61.50
delex	C	63.47	67.67	65.50
	B	**64.74**	**68.15**	**66.40**

Table 2. Cross-language parsing accuracy.

7 Self-Training

Finally, we explored the self-training based domain-adaptation technique of McClosky et al. (2006) in this setting. McClosky et al. trained the Brown parser on one domain of English (WSJ), parsed a large corpus of a second domain (NANTC), trained a new Charniak (non-reranking) parser on WSJ plus the parsed NANTC, and tested the new parser on data from a third domain (Brown Corpus). They observed improvement over baseline in spite of the fact that the large corpus was not in the third domain.

Our setting is similar. We train the Brown parser on Danish treebank and apply it to Swedish Acquis. Then we train new Charniak parser on Danish treebank *and* the parsed Swedish Acquis, and test the parser on the Swedish test data. The hope is that the parser will get lexical context for the structures from the parsed Swedish Acquis.

We did not retrain the reranker on the parsed Acquis, as we found it prohibitively expensive in both time and space. Instead, we created a new Brown parser by combining the new Charniak parser, and the old reranker trained only on Danish.

[6] $F = 2 \times P \times R / (P+R)$

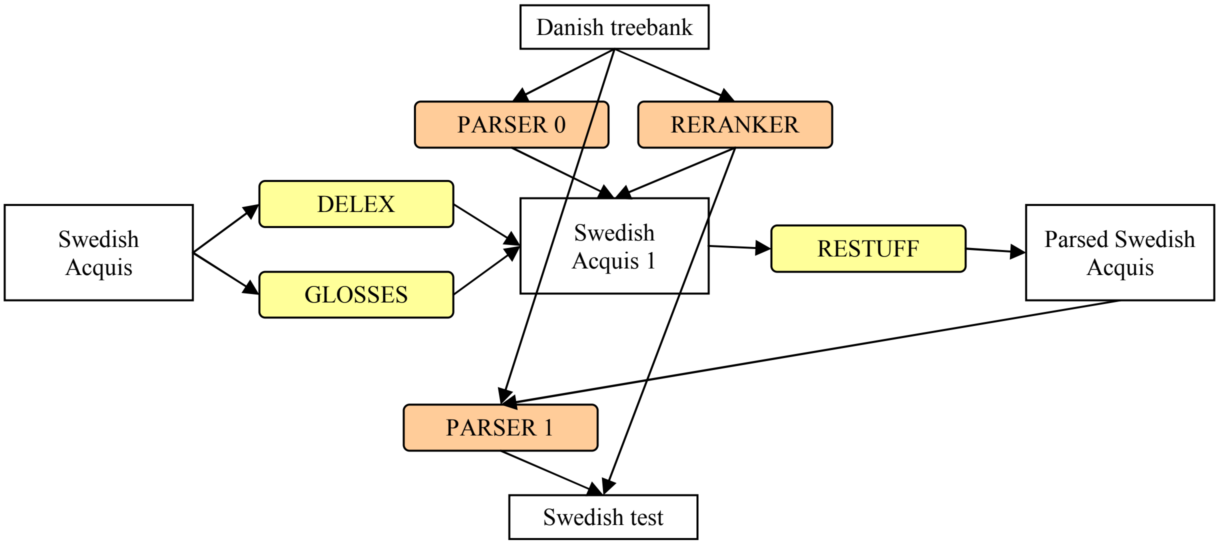

Figure 2. Scheme of the self-training system.

A different scenario is used with the gloss and delex techniques. In this case, we only use delexicalization/glosses to parse the Acquis corpus. The new Charniak model is always trained directly on lexicalized Swedish, i.e. the parsed Acquis is restuffed before being handed over to the trainer. Figure 2 shows the corresponding application chart.

8 Experiments: Part Two

The following table shows the results of the self-training experiments. All F-scores outperform the corresponding results obtained without self-training.

Approach	Parser	P	R	F
plain	C	45.14	43.96	44.54
	B	43.12	42.23	42.67
glosses	C	**62.87**	**66.17**	**64.48**
	B	61.94	64.77	63.32
delex	C	55.87	63.86	59.60
	B	53.87	61.45	57.41

Table 3. Self-training adaptation results.

Not surprisingly, the Danish-trained reranker does not help here. However, even the first-stage parser failed to outperform the Part One results. Therefore the 66.40% labeled F-score of the delexicalized Brown parser is our best result. It improves the baseline by 23% absolute, or 41% error reduction.

9 Discussion

As one way of assessing the usefulness of the result, we compared it to the learning curve on the Swedish treebank. This corresponds to the question "How big a treebank would we have to build, so that the parser trained on the treebank achieves the same F-score?" We measured the F-scores for Swedish-trained parsers on gradually increasing amounts of training data (50, 100, 250, 500, 1000, 2500, 5000 and 10681 sentences).

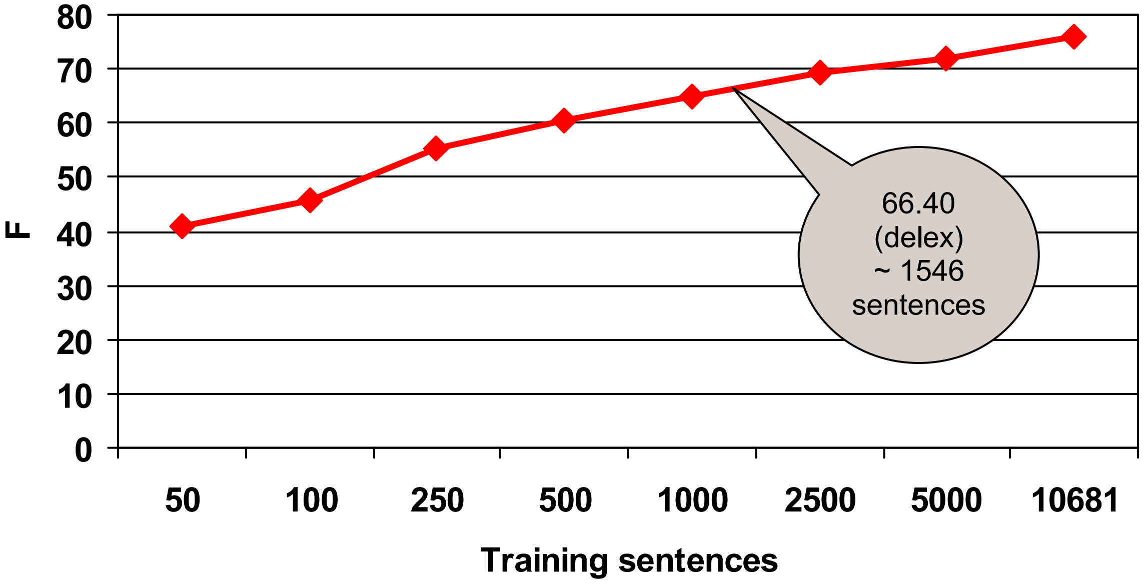

Figure 3. The learning curve on the Swedish training data.

The learning curve is shown in Figure 3. Using interpolation, we see that more than 1500 Swedish parse trees would be required for training, in order to achieve the performance we obtained by adapting an existing Danish treebank. This result is similar in spirit to the results Hwa et al. (2004) report when training a Chinese parser using dependency trees projected from English. As they observe, creating a treebank of even a few thousand trees is a daunting undertaking – consistent annotation typically requires careful design of guidelines for the annotators, testing of the guidelines on data, refinement of those guidelines, ramp-up of annotators, double-annotation for quality control, and so forth. As a case in point, the Prague Dependency Treebank (Böhmová et al, 2003) project began in 1996, and required almost a year for its first 1000 sentences to appear (although things sped up quickly, and over 20000 sentences were available by fall 1998). In contrast, if the source and target language are sufficiently related – consider Danish and Swedish, as we have done, or Hindi and Urdu – our approach should in principle permit a parser to be constructed in a matter of days.

9.1 Ways to Improve: Future Work

The 77.01% F-score of a parser trained on delexicalized automatically assigned hybrid Swedish tags is an upper bound. Some obvious ways of getting closer to it include better treebank and tag-set mapping and better tagging. In addition, we are interested in seeing to what extent performance can be further improved by better iterative self-training.

We also want to explore classifier combination techniques on glosses, delexicalization, and the N-best outputs of the Charniak parser. One could also go further, and explore a combination of techniques, e.g. taking advantage of the ideas proposed here in tandem with unsupervised parsing (as in Bod 2006) or projection of annotations across a parallel corpus (as in Hwa et al. 2004).

Acknowledgements

The authors thank Eugene Charniak and Mark Johnson for making their reranking parser available, as well as the creators of the corpora used in this research. We also thank the anonymous reviewers for useful remarks on where to focus our workshop presentation.

The research reported on in this paper has been supported by the Fulbright-Masaryk Fellowship (first author), and by Grant No. N00014-01-1-0685 ONR. Ongoing research (first author) is supported by the Ministry of Education of the Czech Republic, project MSM0021620838, and Czech Academy of Sciences, project No. 1ET101470416.

References

Rens Bod. 2006a. *Unsupervised Parsing with U-DOP.* In: Proceedings of the Conference on Natural Language Learning (CoNLL-2006). New York, New York, USA.

Rens Bod. 2006b. *An All-Subtrees Approach to Unsupervised Parsing.* In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL (COLING-ACL-2006). Sydney, Australia.

Alena Böhmová, Jan Hajič, Eva Hajičová, Barbora Hladká. 2003. *The Prague Dependency Treebank: A Three-Level Annotation Scenario.* In: Anne Abeillé (ed.): Treebanks: Building and Using Syntactically Annotated Corpora. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Eugene Charniak, Mark Johnson. 2005. *Coarse-to-Fine N-Best Parsing and MaxEnt Discriminative Reranking.* In: Proceedings of the 43rd Annual Meeting of the ACL (ACL-2005), pp. 173–180. Ann Arbor, Michigan, USA.

Michael Collins. 1997. *Three Generative, Lexicalized Models for Statistical Parsing.* In: Proceedings of the 35th Annual Meeting of the ACL, pp. 16–23. Madrid, Spain.

Michael Collins, Jan Hajič, Lance Ramshaw, Christoph Tillmann. 1999. *A Statistical Parser for Czech.* In: Proceedings of the 37th Annual Meeting of the ACL (ACL-1999), pp. 505–512. College Park, Maryland, USA.

Jan Hajič. 2004. *Disambiguation of Rich Inflection (Computational Morphology of Czech).* Karolinum, Charles University Press, Praha, Czechia.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, Okan Kolak. 2004. *Bootstrapping Parsers via Syntactic Projection across Parallel Texts.* In: Natural Language Engineering 1 (1): 1–15. Cambridge University Press, Cambridge, England.

Dan Klein, Christopher D. Manning. 2004. *Corpus-Based Induction of Syntactic Structure: Models of Dependency and Constituency.* In: Proceedings of the 42nd Annual Meeting of the ACL (ACL-2004). Barcelona, Spain.

Philipp Koehn, Franz Josef Och, Daniel Marcu. 2003. Statistical Phrase-Based Translation. In: Proceedings of HLT-NAACL 2003, pp. 127–133. Edmonton, Canada.

Matthias T. Kromann, Line Mikkelsen, Stine Kern Lynge. 2004. *Danish Dependency Treebank.* At: http://www.id.cbs.dk/~mtk/treebank/. København, Denmark.

Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz. 1993. *Building a Large Annotated Corpus of English: the Penn Treebank.* In: Computational Linguistics, vol. 19, pp. 313–330.

David McClosky, Eugene Charniak, Mark Johnson. 2006. *Reranking and Self-Training for Parser Adaptation.* In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL (COLING-ACL-2006). Sydney, Australia.

Joakim Nivre, Jens Nilsson, Johan Hall. 2006. *Talbanken05: A Swedish Treebank with Phrase Structure and Dependency Annotation.* In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-2006). May 24-26. Genova, Italy.

Franz Josef Och, Hermann Ney. 2000. Improved Statistical Alignment Models. In: Proceedings of the 38th Annual Meeting of the ACL (ACL-2000), pp. 440–447. Hong Kong, China.

Philip Resnik, Noah A. Smith. 2003. *The Web as a Parallel Corpus.* In: Computational Linguistics, 29(3), pp. 349–380.

Mark Steedman, Miles Osborne, Anoop Sarkar, Stephen Clark, Rebecca Hwa, Julia Hockenmaier, Paul Ruhlen, Steven Baker, Jeremiah Crim. 2003. *Bootstrapping Statistical Parsers from Small Datasets.* In: Proceedings of the 11th Conference of the European Chapter of the ACL (EACL-2003). Budapest, Hungary.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, Dániel Varga. 2006. *The JRC-Acquis: A Multilingual Aligned Parallel Corpus with 20+ Languages.* In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-2006). May 24-26. Genova, Italy.

Fei Xia, Martha Palmer. 2001. *Converting Dependency Structures to Phrase Structures.* In: Proceedings of the 1st Human Language Technology Conference (HLT-2001). San Diego, California, USA.