The CoNLL 2007 Shared Task on Dependency Parsing

Joakim Nivre[*†] **Johan Hall**[*] **Sandra Kübler**[‡] **Ryan McDonald**[**]
Jens Nilsson[*] **Sebastian Riedel**[††] **Deniz Yuret**[‡‡]

[*]Växjö University, School of Mathematics and Systems Engineering, *first.last*@vxu.se
[†]Uppsala University, Dept. of Linguistics and Philology, joakim.nivre@lingfil.uu.se
[‡]Indiana University, Department of Linguistics, skuebler@indiana.edu
[**]Google Inc., ryanmcd@google.com
[††]University of Edinburgh, School of Informatics, S.R.Riedel@sms.ed.ac.uk
[‡‡]Koç University, Dept. of Computer Engineering, dyuret@ku.edu.tr

Abstract

The Conference on Computational Natural Language Learning features a shared task, in which participants train and test their learning systems on the same data sets. In 2007, as in 2006, the shared task has been devoted to dependency parsing, this year with both a multilingual track and a domain adaptation track. In this paper, we define the tasks of the different tracks and describe how the data sets were created from existing treebanks for ten languages. In addition, we characterize the different approaches of the participating systems, report the test results, and provide a first analysis of these results.

1 Introduction

Previous shared tasks of the Conference on Computational Natural Language Learning (CoNLL) have been devoted to chunking (1999, 2000), clause identification (2001), named entity recognition (2002, 2003), and semantic role labeling (2004, 2005). In 2006 the shared task was multilingual dependency parsing, where participants had to train a single parser on data from thirteen different languages, which enabled a comparison not only of parsing and learning methods, but also of the performance that can be achieved for different languages (Buchholz and Marsi, 2006).

In dependency-based syntactic parsing, the task is to derive a syntactic structure for an input sentence by identifying the syntactic *head* of each word in the sentence. This defines a *dependency graph*, where the nodes are the words of the input sentence and the arcs are the binary relations from head to dependent. Often, but not always, it is assumed that all words except one have a syntactic head, which means that the graph will be a tree with the single independent word as the root. In *labeled* dependency parsing, we additionally require the parser to assign a specific type (or label) to each dependency relation holding between a head word and a dependent word.

In this year's shared task, we continue to explore data-driven methods for multilingual dependency parsing, but we add a new dimension by also introducing the problem of domain adaptation. The way this was done was by having two separate tracks: a multilingual track using essentially the same setup as last year, but with partly different languages, and a domain adaptation track, where the task was to use machine learning to adapt a parser for a single language to a new domain. In total, test results were submitted for twenty-three systems in the multilingual track, and ten systems in the domain adaptation track (six of which also participated in the multilingual track). Not everyone submitted papers describing their system, and some papers describe more than one system (or the same system in both tracks), which explains why there are only (!) twenty-one papers in the proceedings.

In this paper, we provide task definitions for the two tracks (section 2), describe data sets extracted from available treebanks (section 3), report results for all systems in both tracks (section 4), give an overview of approaches used (section 5), provide a first analysis of the results (section 6), and conclude with some future directions (section 7).

Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007, pp. 915–932,
Prague, June 2007.

2 Task Definition

In this section, we provide the task definitions that were used in the two tracks of the CoNLL 2007 Shard Task, the multilingual track and the domain adaptation track, together with some background and motivation for the design choices made. First of all, we give a brief description of the data format and evaluation metrics, which were common to the two tracks.

2.1 Data Format and Evaluation Metrics

The data sets derived from the original treebanks (section 3) were in the same column-based format as for the 2006 shared task (Buchholz and Marsi, 2006). In this format, sentences are separated by a blank line; a sentence consists of one or more tokens, each one starting on a new line; and a token consists of the following ten fields, separated by a single tab character:

1. ID: Token counter, starting at 1 for each new sentence.

2. FORM: Word form or punctuation symbol.

3. LEMMA: Lemma or stem of word form, or an underscore if not available.

4. CPOSTAG: Coarse-grained part-of-speech tag, where the tagset depends on the language.

5. POSTAG: Fine-grained part-of-speech tag, where the tagset depends on the language, or identical to the coarse-grained part-of-speech tag if not available.

6. FEATS: Unordered set of syntactic and/or morphological features (depending on the particular language), separated by a vertical bar (|), or an underscore if not available.

7. HEAD: Head of the current token, which is either a value of ID or zero (0). Note that, depending on the original treebank annotation, there may be multiple tokens with HEAD=0.

8. DEPREL: Dependency relation to the HEAD. The set of dependency relations depends on the particular language. Note that, depending on the original treebank annotation, the dependency relation when HEAD=0 may be meaningful or simply ROOT.

9. PHEAD: Projective head of current token, which is either a value of ID or zero (0), or an underscore if not available.

10. PDEPREL: Dependency relation to the PHEAD, or an underscore if not available.

The PHEAD and PDEPREL were not used at all in this year's data sets (i.e., they always contained underscores) but were maintained for compatibility with last year's data sets. This means that, in practice, the first six columns can be considered as *input* to the parser, while the HEAD and DEPREL fields are the *output* to be produced by the parser. Labeled training sets contained all ten columns; blind test sets only contained the first six columns; and gold standard test sets (released only after the end of the test period) again contained all ten columns. All data files were encoded in UTF-8.

The official evaluation metric in both tracks was the *labeled attachment score* (LAS), i.e., the percentage of tokens for which a system has predicted the correct HEAD and DEPREL, but results were also reported for *unlabeled attachment score* (UAS), i.e., the percentage of tokens with correct HEAD, and the *label accuracy* (LA), i.e., the percentage of tokens with correct DEPREL. One important difference compared to the 2006 shared task is that all tokens were counted as "scoring tokens", including in particular all punctuation tokens. The official evaluation script, eval07.pl, is available from the shared task website.[1]

2.2 Multilingual Track

The multilingual track of the shared task was organized in the same way as the 2006 task, with annotated training and test data from a wide range of languages to be processed with one and the same parsing system. This system must therefore be able to learn from training data, to generalize to unseen test data, and to handle multiple languages, possibly by adjusting a number of hyper-parameters. Participants in the multilingual track were expected to submit parsing results for all languages involved.

[1] http://depparse.uvt.nl/depparse-wiki/SoftwarePage

One of the claimed advantages of dependency parsing, as opposed to parsing based on constituent analysis, is that it extends naturally to languages with free or flexible word order. This explains the interest in recent years for multilingual evaluation of dependency parsers. Even before the 2006 shared task, the parsers of Collins (1997) and Charniak (2000), originally developed for English, had been adapted for dependency parsing of Czech, and the parsing methodology proposed by Kudo and Matsumoto (2002) and Yamada and Matsumoto (2003) had been evaluated on both Japanese and English. The parser of McDonald and Pereira (2006) had been applied to English, Czech and Danish, and the parser of Nivre et al. (2007) to ten different languages. But by far the largest evaluation of multilingual dependency parsing systems so far was the 2006 shared task, where nineteen systems were evaluated on data from thirteen languages (Buchholz and Marsi, 2006).

One of the conclusions from the 2006 shared task was that parsing accuracy differed greatly between languages and that a deeper analysis of the factors involved in this variation was an important problem for future research. In order to provide an extended empirical foundation for such research, we tried to select the languages and data sets for this year's task based on the following desiderata:

- The selection of languages should be typologically varied and include both new languages and old languages (compared to 2006).

- The creation of the data sets should involve as little conversion as possible from the original treebank annotation, meaning that preference should be given to treebanks with dependency annotation.

- The training data sets should include at least 50,000 tokens and at most 500,000 tokens.[2]

The final selection included data from Arabic, Basque, Catalan, Chinese, Czech, English, Greek, Hungarian, Italian, and Turkish. The treebanks from which the data sets were extracted are described in section 3.

2.3 Domain Adaptation Track

One well known characteristic of data-driven parsing systems is that they typically perform much worse on data that does not come from the training domain (Gildea, 2001). Due to the large overhead in annotating text with deep syntactic parse trees, the need to adapt parsers from domains with plentiful resources (e.g., news) to domains with little resources is an important problem. This problem is commonly referred to as *domain adaptation*, where the goal is to adapt annotated resources from a *source domain* to a *target domain* of interest.

Almost all prior work on domain adaptation assumes one of two scenarios. In the first scenario, there are limited annotated resources available in the target domain, and many studies have shown that this may lead to substantial improvements. This includes the work of Roark and Bacchiani (2003), Florian et al. (2004), Chelba and Acero (2004), Daumé and Marcu (2006), and Titov and Henderson (2006). Of these, Roark and Bacchiani (2003) and Titov and Henderson (2006) deal specifically with syntactic parsing. The second scenario assumes that there are no annotated resources in the target domain. This is a more realistic situation and is considerably more difficult. Recent work by McClosky et al. (2006) and Blitzer et al. (2006) have shown that the existence of a large unlabeled corpus in the new domain can be leveraged in adaptation. For this shared-task, we are assuming the latter setting – *no annotated resources in the target domain*.

Obtaining adequate annotated syntactic resources for multiple languages is already a challenging problem, which is only exacerbated when these resources must be drawn from multiple and diverse domains. As a result, the only language that could be feasibly tested in the domain adaptation track was English.

The setup for the domain adaptation track was as follows. Participants were provided with a large annotated corpus from the source domain, in this case sentences from the Wall Street Journal. Participants were also provided with data from three different target domains: biomedical abstracts (development data), chemical abstracts (test data 1), and parent-child dialogues (test data 2). Additionally, a large

[2] The reason for having an upper bound on the training set size was the fact that, in 2006, some participants could not train on all the data for some languages because of time limitations. Similar considerations also led to the decision to have a smaller number of languages this year (ten, as opposed to thirteen).

unlabeled corpus for each data set (training, development, test) was provided. The goal of the task was to use the annotated source data, plus any unlabeled data, to produce a parser that is accurate for each of the test sets from the target domains.[3]

Participants could submit systems in either the "open" or "closed" class (or both). The closed class requires a system to use only those resources provided as part of the shared task. The open class allows a system to use additional resources provided those resources are not drawn from the same domain as the development or test sets. An example might be a part-of-speech tagger trained on the entire Penn Treebank and not just the subset provided as training data, or a parser that has been hand-crafted or trained on a different training set.

3 Treebanks

In this section, we describe the treebanks used in the shared task and give relevant information about the data sets created from them.

3.1 Multilingual Track

Arabic The analytical syntactic annotation of the Prague Arabic Dependency Treebank (PADT) (Hajič et al., 2004) can be considered a pure dependency annotation. The conversion, done by Otakar Smrz, from the original format to the column-based format described in section 2.1 was therefore relatively straightforward, although not all the information in the original annotation could be transfered to the new format. PADT was one of the treebanks used in the 2006 shared task but then only contained about 54,000 tokens. Since then, the size of the treebank has more than doubled, with around 112,000 tokens. In addition, the morphological annotation has been made more informative. It is also worth noting that the parsing units in this treebank are in many cases larger than conventional sentences, which partly explains the high average number of tokens per "sentence" (Buchholz and Marsi, 2006).

[3] Note that annotated development data for the target domain was only provided for the development domain, biomedical abstracts. For the two test domains, chemical abstracts and parent-child dialogues, the only annotated data sets were the gold standard test sets, released only after test runs had been submitted.

Basque For Basque, we used the 3LB Basque treebank (Aduriz et al., 2003). At present, the treebank consists of approximately 3,700 sentences, 334 of which were used as test data. The treebank comprises literary and newspaper texts. It is annotated in a dependency format and was converted to the CoNLL format by a team led by Koldo Gojenola.

Catalan The Catalan section of the CESS-ECE Syntactically and Semantically Annotated Corpora (Martí et al., 2007) is annotated with, among other things, constituent structure and grammatical functions. A head percolation table was used for automatically converting the constituent trees into dependency trees. The original data only contains functions related to the verb, and a function table was used for deriving the remaining syntactic functions. The conversion was performed by a team led by Lluís Màrquez and Antònia Martí.

Chinese The Chinese data are taken from the Sinica treebank (Chen et al., 2003), which contains both syntactic functions and semantic functions. The syntactic head was used in the conversion to the CoNLL format, carried out by Yu-Ming Hsieh and the organizers of the 2006 shared task, and the syntactic functions were used wherever it was possible. The training data used is basically the same as for the 2006 shared task, except for a few corrections, but the test data is new for this year's shared task. It is worth noting that the parsing units in this treebank are sometimes smaller than conventional sentence units, which partly explains the low average number of tokens per "sentence" (Buchholz and Marsi, 2006).

Czech The analytical syntactic annotation of the Prague Dependency Treebank (PDT) (Böhmová et al., 2003) is a pure dependency annotation, just as for PADT. It was also used in the shared task 2006, but there are two important changes compared to last year. First, version 2.0 of PDT was used instead of version 1.0, and a conversion script was created by Zdenek Zabokrtsky, using the new XML-based format of PDT 2.0. Secondly, due to the upper bound on training set size, only sections 1–3 of PDT constitute the training data, which amounts to some 450,000 tokens. The test data is a small subset of the development test set of PDT.

English For English we used the Wall Street Journal section of the Penn Treebank (Marcus et al., 1993). In particular, we used sections 2-11 for training and a subset of section 23 for testing. As a preprocessing stage we removed many functions tags from the non-terminals in the phrase structure representation to make the representations more uniform with out-of-domain test sets for the domain adaptation track (see section 3.2). The resulting data set was then converted to dependency structures using the procedure described in Johansson and Nugues (2007a). This work was done by Ryan McDonald.

Greek The Greek Dependency Treebank (GDT) (Prokopidis et al., 2005) adopts a dependency structure annotation very similar to those of PDT and PADT, which means that the conversion by Prokopis Prokopidis was relatively straightforward. GDT is one of the smallest treebanks in this year's shared task (about 65,000 tokens) and contains sentences of Modern Greek. Just like PDT and PADT, the treebank contains more than one level of annotation, but we only used the analytical level of GDT.

Hungarian For the Hungarian data, the Szeged treebank (Csendes et al., 2005) was used. The treebank is based on texts from six different genres, ranging from legal newspaper texts to fiction. The original annotation scheme is constituent-based, following generative principles. It was converted into dependencies by Zóltan Alexin based on heuristics.

Italian The data set used for Italian is a subset of the balanced section of the Italian Syntactic-Semantic Treebank (ISST) (Montemagni et al., 2003) and consists of texts from the newspaper *Corriere della Sera* and from periodicals. A team led by Giuseppe Attardi, Simonetta Montemagni, and Maria Simi converted the annotation to the CoNLL format, using information from two different annotation levels, the constituent structure level and the dependency structure level.

Turkish For Turkish we used the METU-Sabancı Turkish Treebank (Oflazer et al., 2003), which was also used in the 2006 shared task. A new test set of about 9,000 tokens was provided by Gülşen Eryiğit (Eryiğit, 2007), who also handled the conversion to the CoNLL format, which means that we could use all the approximately 65,000 tokens of the original treebank for training. The rich morphology of Turkish requires the basic tokens in parsing to be inflectional groups (IGs) rather than words. IGs of a single word are connected to each other deterministically using dependency links labeled DERIV, referred to as word-internal dependencies in the following, and the FORM and the LEMMA fields may be empty (they contain underscore characters in the data files). Sentences do not necessarily have a unique root; most internal punctuation and a few foreign words also have HEAD=0.

3.2 Domain Adaptation Track

As mentioned previously, the source data is drawn from a corpus of news, specifically the Wall Street Journal section of the Penn Treebank (Marcus et al., 1993). This data set is identical to the English training set from the multilingual track (see section 3.1).

For the target domains we used three different labeled data sets. The first two were annotated as part of the PennBioIE project (Kulick et al., 2004) and consist of sentences drawn from either biomedical or chemical research abstracts. Like the source WSJ corpus, this data is annotated using the Penn Treebank phrase structure scheme. To convert these sets to dependency structures we used the same procedure as before (Johansson and Nugues, 2007a). Additional care was taken to remove sentences that contained non-WSJ part-of-speech tags or non-terminals (e.g., HYPH part-of-speech tag indicating a hyphen). Furthermore, the annotation scheme for gaps and traces was made consistent with the Penn Treebank wherever possible. As already mentioned, the biomedical data set was distributed as a development set for the training phase, while the chemical data set was only used for final testing.

The third target data set was taken from the CHILDES database (MacWhinney, 2000), in particular the EVE corpus (Brown, 1973), which has been annotated with dependency structures. Unfortunately the dependency labels of the CHILDES data were inconsistent with those of the WSJ, biomedical and chemical data sets, and we therefore opted to only evaluate unlabeled accuracy for this data set. Furthermore, there was an inconsistency in how main and auxiliary verbs were annotated for this data set relative to others. As a result of this, submitting

Multilingual											Domain adaptation	
	Ar	Ba	Ca	Ch	Cz	En	Gr	Hu	It	Tu	PCHEM	CHILDES
Language family	Sem.	Isol.	Rom.	Sin.	Sla.	Ger.	Hel.	F.-U.	Rom.	Tur.	Ger.	
Annotation	d	d	c+f	c+f	d	c+f	d	c+f	c+f	d	c+f	d
Training data											Development data	
Tokens (k)	112	51	431	337	432	447	65	132	71	65	5	
Sentences (k)	2.9	3.2	15.0	57.0	25.4	18.6	2.7	6.0	3.1	5.6	0.2	
Tokens/sentence	38.3	15.8	28.8	5.9	17.0	24.0	24.2	21.8	22.9	11.6	25.1	
LEMMA	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes	Yes	No	
No. CPOSTAG	15	25	17	13	12	31	18	16	14	14	25	
No. POSTAG	21	64	54	294	59	45	38	43	28	31	37	
No. FEATS	21	359	33	0	71	0	31	50	21	78	0	
No. DEPREL	29	35	42	69	46	20	46	49	22	25	18	
No. DEPREL H=0	18	17	1	1	8	1	22	1	1	1	1	
% HEAD=0	8.7	9.7	3.5	16.9	11.6	4.2	8.3	4.6	5.4	12.8	4.0	
% HEAD left	79.2	44.5	60.0	24.7	46.9	49.0	44.8	27.4	65.0	3.8	50.0	
% HEAD right	12.1	45.8	36.5	58.4	41.5	46.9	46.9	68.0	29.6	83.4	46.0	
HEAD=0/sentence	3.3	1.5	1.0	1.0	2.0	1.0	2.0	1.0	1.2	1.5	1.0	
% Non-proj. arcs	0.4	2.9	0.1	0.0	1.9	0.3	1.1	2.9	0.5	5.5	0.4	
% Non-proj. sent.	10.1	26.2	2.9	0.0	23.2	6.7	20.3	26.4	7.4	33.3	8.0	
Punc. attached	S	S	A	S	S	A	S	A	A	S	A	
DEPRELS for punc.	10	13	6	29	16	13	15	1	10	12	8	
Test data											PCHEM	CHILDES
Tokens	5124	5390	5016	5161	4724	5003	4804	7344	5096	4513	5001	4999
Sentences	131	334	167	690	286	214	197	390	249	300	195	666
Tokens/sentence	39.1	16.1	30.0	7.5	16.5	23.4	24.4	18.8	20.5	15.0	25.6	12.9
% New words	12.44	24.98	4.35	9.70	12.58	3.13	12.43	26.10	15.07	36.29	31.33	6.10
% New lemmas	2.82	11.13	3.36	n/a	5.28	n/a	5.82	14.80	8.24	9.95	n/a	n/a

Table 1: Characteristics of the data sets for the 10 languages of the multilingual track and the development set and the two test sets of the domain adaptation track.

results for the CHILDES data was considered optional. Like the chemical data set, this data set was only used for final testing.

Finally, a large corpus of unlabeled in-domain data was provided for each data set and made available for training. This data was drawn from the WSJ, PubMed.com (specific to biomedical and chemical research literature), and the CHILDES data base. The data was tokenized to be as consistent as possible with the WSJ training set.

3.3 Overview

Table 1 describes the characteristics of the data sets. For the multilingual track, we provide statistics over the training and test sets; for the domain adaptation track, the statistics were extracted from the development set. Following last year's shared task practice (Buchholz and Marsi, 2006), we use the following definition of projectivity: An arc (i, j) is projective iff all nodes occurring between i and j are dominated by i (where dominates is the transitive closure of the arc relation).

In the table, the languages are abbreviated to their first two letters. Language families are: Semitic, Isolate, Romance, Sino-Tibetan, Slavic, Germanic, Hellenic, Finno-Ugric, and Turkic. The type of the original annotation is either constituents plus (some) functions (c+f) or dependencies (d). For the training data, the number of words and sentences are given in multiples of thousands, and the average length of a sentence in words (including punctuation tokens). The following rows contain information about whether lemmas are available, the number of coarse- and fine-grained part-of-speech tags, the number of feature components, and the number of dependency labels. Then information is given on how many different dependency labels can co-occur with HEAD=0, the percentage of HEAD=0 dependencies, and the percentage of heads preceding (left) or succeeding (right) a token (giving an indication of whether a language is predominantly head-initial or head-final). This is followed by the average number of HEAD=0 dependencies per sentence and the percentage of non-projective arcs and sentences. The last two rows show whether punctuation tokens are attached as dependents of other tokens (A=Always, S=Sometimes) and specify the number of dependency labels that exist for punctuation tokens. Note that punctuation is defined as any token belonging to the UTF-8 category of punctuation. This means, for example, that any token having an underscore in the FORM field (which happens for word-internal IGs in Turkish) is also counted as punctuation here.

For the test sets, the number of words and sentences as well as the ratio of words per sentence are listed, followed by the percentage of new words and lemmas (if applicable). For the domain adaptation sets, the percentage of new words is computed with regard to the training set (Penn Treebank).

4 Submissions and Results

As already stated in the introduction, test runs were submitted for twenty-three systems in the multilingual track, and ten systems in the domain adaptation track (six of which also participated in the multilingual track). In the result tables below, systems are identified by the last name of the team member listed first when test runs were uploaded for evaluation. In general, this name is also the first author of a paper describing the system in the proceedings, but there are a few exceptions and complications. First of all, for four out of twenty-seven systems, no paper was submitted to the proceedings. This is the case for the systems of Jia, Maes et al., Nash, and Zeman, which is indicated by the fact that these names appear in italics in all result tables. Secondly, two teams submitted two systems each, which are described in a single paper by each team. Thus, the systems called "Nilsson" and "Hall, J." are both described in Hall et al. (2007a), while the systems called "Duan (1)" and "Duan (2)" are both described in Duan et al. (2007). Finally, please pay attention to the fact that there are two teams, where the first author's last name is Hall. Therefore, we use "Hall, J." and "Hall, K.", to disambiguate between the teams involving Johan Hall (Hall et al., 2007a) and Keith Hall (Hall et al., 2007b), respectively.

Tables 2 and 3 give the scores for the multilingual track in the CoNLL 2007 shared task. The Average column contains the average score for all ten languages, which determines the ranking in this track. Table 4 presents the results for the domain adaptation track, where the ranking is determined based on the PCHEM results only, since the CHILDES data set was optional. Note also that there are no labeled

Team	Average	Arabic	Basque	Catalan	Chinese	Czech	English	Greek	Hungarian	Italian	Turkish
Nilsson	80.32(1)	76.52(1)	76.94(1)	88.70(1)	75.82(15)	77.98(3)	88.11(5)	74.65(2)	80.27(1)	84.40(1)	79.79(2)
Nakagawa	80.29(2)	75.08(2)	72.56(7)	87.90(3)	83.84(2)	80.19(1)	88.41(3)	76.31(1)	76.74(8)	83.61(3)	78.22(5)
Titov	79.90(3)	74.12(6)	75.49(3)	87.40(6)	82.14(7)	77.94(4)	88.39(4)	73.52(10)	77.94(4)	82.26(6)	79.81(1)
Sagae	79.90(4)	74.71(4)	74.64(6)	88.16(2)	84.69(1)	74.83(8)	89.01(2)	73.58(8)	79.53(2)	83.91(2)	75.91(10)
Hall, J.	79.80(5)*	74.75(3)	74.99(5)	87.74(4)	83.51(3)	77.22(6)	85.81(12)	74.21(6)	78.09(3)	82.48(5)	79.24(3)
Carreras	79.09(6)*	70.20(11)	75.75(2)	87.60(5)	80.86(10)	78.60(2)	89.61(1)	73.56(9)	75.42(9)	83.46(4)	75.85(11)
Attardi	78.27(7)	72.66(8)	69.48(12)	86.86(7)	81.50(8)	77.37(5)	85.85(10)	73.92(7)	76.81(7)	81.34(8)	76.87(7)
Chen	78.06(8)	74.65(5)	72.39(8)	86.66(8)	81.24(9)	73.69(10)	83.81(13)	74.42(3)	75.34(10)	82.04(7)	76.31(9)
Duan (1)	77.70(9)*	69.91(13)	71.26(9)	84.95(10)	82.58(6)	75.34(7)	85.83(11)	74.29(4)	77.06(5)	80.75(9)	75.03(12)
Hall, K.	76.91(10)*	73.40(7)	69.81(11)	82.38(14)	82.77(4)	72.27(12)	81.93(15)	74.21(5)	74.20(11)	80.69(10)	77.42(6)
Schiehlen	76.18(11)	70.08(12)	66.77(14)	85.75(9)	80.04(11)	73.86(9)	86.21(9)	72.29(12)	73.90(12)	80.46(11)	72.48(15)
Johansson	75.78(12)*	71.76(9)	75.08(4)	83.33(12)	76.30(14)	70.98(13)	80.29(17)	72.77(11)	71.31(13)	77.55(14)	78.46(4)
Mannem	74.54(13)*	71.55(10)	65.64(15)	84.47(11)	73.76(17)	70.68(14)	81.55(16)	71.69(13)	70.94(14)	78.67(13)	76.42(8)
Wu	73.02(14)*	66.16(14)	70.71(10)	81.44(15)	74.69(16)	66.72(16)	79.49(18)	70.63(14)	69.08(15)	78.79(12)	72.52(14)
Nguyen	72.53(15)*	63.58(16)	58.18(17)	83.23(13)	79.77(12)	72.54(11)	86.73(6)	70.42(15)	68.12(17)	75.06(16)	67.63(17)
Maes	70.66(16)*	65.12(15)	69.05(13)	79.21(16)	70.97(18)	67.38(15)	69.68(21)	68.59(16)	68.93(16)	73.63(18)	74.03(13)
Canisius	66.99(17)*	59.13(18)	63.17(16)	75.44(17)	70.45(19)	56.14(17)	77.27(19)	60.35(18)	64.31(19)	75.57(15)	68.09(16)
Jia	63.00(18)*	63.37(17)	57.61(18)	23.35(20)	76.36(13)	54.95(18)	82.93(14)	65.45(17)	66.61(18)	74.65(17)	64.68(18)
Zeman	54.87(19)	46.06(20)	50.61(20)	62.94(19)	54.49(20)	50.21(20)	53.59(22)	55.29(19)	55.24(20)	62.13(19)	58.10(19)
Marinov	54.55(20)*	54.00(19)	51.24(19)	69.42(18)	49.87(21)	53.47(19)	52.11(23)	54.33(20)	44.47(21)	59.75(20)	56.88(20)
Duan (2)	24.62(21)*				82.64(5)		86.69(7)		76.89(6)		
Nash	8.65(22)*						86.49(8)				
Shimizu	7.20(23)						72.02(20)				

Table 2: Labeled attachment score (LAS) for the multilingual track in the CoNLL 2007 shared task. Teams are denoted by the last name of their first member, with italics indicating that there is no corresponding paper in the proceedings. The number in parentheses next to each score gives the rank. A star next to a score in the Average column indicates a statistically significant difference with the next lower rank.

Team	Average	Arabic	Basque	Catalan	Chinese	Czech	English	Greek	Hungarian	Italian	Turkish
Nakagawa	86.55(1)*	86.09(1)	81.04(5)	92.86(4)	88.88(2)	86.28(1)	90.13(2)	84.08(1)	82.49(3)	87.91(1)	85.77(3)
Nilsson	85.71(2)	85.81(2)	82.84(1)	93.12(3)	84.52(12)	83.59(4)	88.93(5)	81.22(4)	83.55(1)	87.77(2)	85.77(2)
Titov	85.62(3)	83.18(7)	81.93(2)	93.40(1)	87.91(4)	84.19(3)	89.73(4)	81.20(5)	82.18(4)	86.26(6)	86.22(1)
Sagae	85.29(4)*	84.04(4)	81.19(3)	93.34(2)	88.94(1)	81.27(8)	89.87(3)	80.37(11)	83.51(2)	87.68(3)	82.72(9)
Carreras	84.79(5)	81.48(10)	81.11(4)	92.46(5)	86.20(9)	85.16(2)	90.63(1)	81.37(3)	79.92(9)	87.19(4)	82.41(10)
Hall, J.	84.74(6)*	84.21(3)	80.61(6)	92.20(6)	87.60(5)	82.35(6)	86.77(12)	80.66(9)	81.71(6)	86.26(5)	85.04(5)
Attardi	83.96(7)*	82.53(8)	76.88(11)	91.41(7)	86.73(8)	83.40(5)	86.99(10)	80.75(8)	81.81(5)	85.54(8)	83.56(7)
Chen	83.22(8)	83.49(5)	78.65(8)	90.87(8)	85.91(10)	80.14(11)	84.91(13)	81.16(6)	79.25(11)	85.91(7)	81.92(12)
Hall, K.	83.08(9)	83.45(6)	78.55(9)	87.80(15)	87.91(3)	78.47(12)	83.21(15)	82.04(2)	79.34(10)	84.81(9)	85.18(4)
Duan (1)	82.77(10)	79.04(13)	77.59(10)	89.71(12)	86.88(7)	80.82(10)	86.97(11)	80.77(7)	80.66(7)	84.20(11)	81.03(13)
Schiehlen	82.42(11)*	81.07(11)	73.30(14)	90.79(10)	85.45(11)	81.73(7)	88.91(6)	80.47(10)	78.61(12)	84.54(10)	79.33(15)
Johansson	81.13(12)*	80.91(12)	80.43(7)	88.34(13)	81.30(15)	77.39(13)	81.43(18)	79.58(12)	75.53(15)	81.55(15)	84.80(6)
Mannem	80.30(13)	81.56(9)	72.88(15)	89.81(11)	78.84(17)	77.20(14)	82.81(16)	78.89(13)	75.39(16)	82.91(12)	82.74(8)
Nguyen	80.00(14)*	73.46(18)	69.15(18)	88.12(14)	84.05(13)	80.91(9)	88.01(7)	77.56(15)	78.13(13)	80.40(16)	80.19(14)
Jia	78.46(15)	74.20(17)	70.24(16)	90.83(9)	83.39(14)	70.41(18)	84.37(14)	75.65(16)	77.19(14)	82.36(14)	75.96(17)
Wu	78.44(16)*	77.05(14)	75.77(12)	85.85(16)	79.71(16)	73.07(16)	81.69(17)	78.12(14)	72.39(18)	82.57(13)	78.15(16)
Maes	76.60(17)*	75.47(16)	75.27(13)	84.35(17)	76.57(18)	74.03(15)	71.62(21)	75.19(17)	72.93(17)	78.32(18)	82.21(11)
Canisius	74.83(18)*	76.89(15)	70.17(17)	81.64(18)	74.81(19)	72.12(17)	78.23(19)	72.46(18)	67.80(19)	79.08(17)	75.14(18)
Zeman	62.02(19)*	58.55(20)	57.42(20)	68.50(20)	62.93(20)	59.19(20)	58.33(22)	62.89(19)	59.78(20)	68.27(19)	64.30(19)
Marinov	60.83(20)*	64.27(19)	58.55(19)	74.22(19)	56.09(21)	59.57(19)	54.33(23)	61.18(20)	50.39(21)	65.52(20)	64.13(20)
Duan (2)	25.53(21)*				86.94(6)		87.87(8)		80.53(8)		
Nash	8.77(22)*						87.71(9)				
Shimizu	7.79(23)						77.91(20)				

Table 3: Unlabeled attachment scores (UAS) for the multilingual track in the CoNLL 2007 shared task. Teams are denoted by the last name of their first member, with italics indicating that there is no corresponding paper in the proceedings. The number in parentheses next to each score gives the rank. A star next to a score in the Average column indicates a statistically significant difference with the next lower rank.

	LAS		UAS			
Team	PCHEM-c	PCHEM-o	PCHEM-c	PCHEM-o	CHILDES-c	CHILDES-o
Sagae	81.06(1)		83.42(1)			
Attardi	80.40(2)		83.08(3)		58.67(3)	
Dredze	80.22(3)		83.38(2)		61.37(1)	
Nguyen	79.50(4)*		82.04(4)*			
Jia	76.48(5)*		78.92(5)*		57.43(5)	
Bick	71.81(6)*	78.48(1)*	74.71(6)*	81.62(1)*	58.07(4)	62.49(1)
Shimizu	64.15(7)*	63.49(2)	71.25(7)*	70.01(2)*		
Zeman	50.61(8)		54.57(8)		58.89(2)	
Schneider		63.01(3)*		66.53(3)*		60.27(2)
Watson		55.47(4)		62.79(4)		45.61(3)
Wu					52.89(6)	

Table 4: Labeled (LAS) and unlabeled (UAS) attachment scores for the closed (-c) and open (-o) classes of the domain adaptation track in the CoNLL 2007 shared task. Teams are denoted by the last name of their first member, with italics indicating that there is no corresponding paper in the proceedings. The number in parentheses next to each score gives the rank. A star next to a score in the PCHEM columns indicates a statistically significant difference with the next lower rank.

attachment scores for the CHILDES data set, for reasons explained in section 3.2. The number in parentheses next to each score gives the rank. A star next to a score indicates that the difference with the next lower rank is significant at the 5% level using a z-test for proportions. A more complete presentation of the results, including the significance results for all the tasks and their p-values, can be found on the shared task website.[4]

Looking first at the results in the multilingual track, we note that there are a number of systems performing at almost the same level at the top of the ranking. For the average labeled attachment score, the difference between the top score (Nilsson) and the fifth score (Hall, J.) is no more than half a percentage point, and there are generally very few significant differences among the five or six best systems, regardless of whether we consider labeled or unlabeled attachment score. For the closed class of the domain adaptation track, we see a very similar pattern, with the top system (Sagae) being followed very closely by two other systems. For the open class, the results are more spread out, but then there are very few results in this class. It is also worth noting that the top scores in the closed class, somewhat unexpectedly, are higher than the top scores in the open class. But before we proceed to a more detailed analysis of the results (section 6), we will make an attempt to characterize the approaches represented by the different systems.

5 Approaches

In this section we give an overview of the models, inference methods, and learning methods used in the participating systems. For obvious reasons the discussion is limited to systems that are described by a paper in the proceedings. But instead of describing the systems one by one, we focus on the basic methodological building blocks that are often found in several systems although in different combinations. For descriptions of the individual systems, we refer to the respective papers in the proceedings.

Section 5.1 is devoted to system architectures. We then describe the two main paradigms for learning and inference, in this year's shared task as well as in last year's, which we call *transition-based* parsers (section 5.2) and *graph-based* parsers (section 5.3), adopting the terminology of McDonald and Nivre (2007).[5] Finally, we give an overview of the domain adaptation methods that were used (section 5.4).

[4] http://nextens.uvt.nl/depparse-wiki/AllScores

[5] This distinction roughly corresponds to the distinction made by Buchholz and Marsi (2006) between ”stepwise” and “all-pairs” approaches.

5.1 Architectures

Most systems perform some amount of pre- and post-processing, making the actual parsing component part of a sequential workflow of varying length and complexity. For example, most transition-based parsers can only build projective dependency graphs. For languages with non-projective dependencies, graphs therefore need to be projectivized for training and deprojectivized for testing (Hall et al., 2007a; Johansson and Nugues, 2007b; Titov and Henderson, 2007).

Instead of assigning HEAD and DEPREL in a single step, some systems use a two-stage approach for attaching and labeling dependencies (Chen et al., 2007; Dredze et al., 2007). In the first step unlabeled dependencies are generated, in the second step these are labeled. This is particularly helpful for factored parsing models, in which label decisions cannot be easily conditioned on larger parts of the structure due to the increased complexity of inference. One system (Hall et al., 2007b) extends this two-stage approach to a three-stage architecture where the parser and labeler generate an n-best list of parses which in turn is reranked.[6]

In ensemble-based systems several base parsers provide parsing decisions, which are added together for a combined score for each potential dependency arc. The tree that maximizes the sum of these combined scores is taken as the final output parse. This technique is used by Sagae and Tsujii (2007) and in the Nilsson system (Hall et al., 2007a). It is worth noting that both these systems combine transition-based base parsers with a graph-based method for parser combination, as first described by Sagae and Lavie (2006).

Data-driven grammar-based parsers, such as Bick (2007), Schneider et al. (2007), and Watson and Briscoe (2007), need pre- and post-processing in order to map the dependency graphs provided as training data to a format compatible with the grammar used, and vice versa.

5.2 Transition-Based Parsers

Transition-based parsers build dependency graphs by performing sequences of actions, or transitions. Both learning and inference is conceptualized in terms of predicting the correct transition based on the current parser state and/or history. We can further subclassify parsers with respect to the model (or transition system) they adopt, the inference method they use, and the learning method they employ.

5.2.1 Models

The most common model for transition-based parsers is one inspired by shift-reduce parsing, where a parser state contains a stack of partially processed tokens and a queue of remaining input tokens, and where transitions add dependency arcs and perform stack and queue operations. This type of model is used by the majority of transition-based parsers (Attardi et al., 2007; Duan et al., 2007; Hall et al., 2007a; Johansson and Nugues, 2007b; Mannem, 2007; Titov and Henderson, 2007; Wu et al., 2007). Sometimes it is combined with an explicit probability model for transition sequences, which may be conditional (Duan et al., 2007) or generative (Titov and Henderson, 2007).

An alternative model is based on the list-based parsing algorithm described by Covington (2001), which iterates over the input tokens in a sequential manner and evaluates for each preceding token whether it can be linked to the current token or not. This model is used by Marinov (2007) and in component parsers of the Nilsson ensemble system (Hall et al., 2007a). Finally, two systems use models based on LR parsing (Sagae and Tsujii, 2007; Watson and Briscoe, 2007).

5.2.2 Inference

The most common inference technique in transition-based dependency parsing is greedy deterministic search, guided by a classifier for predicting the next transition given the current parser state and history, processing the tokens of the sentence in sequential left-to-right order[7] (Hall et al., 2007a; Mannem, 2007; Marinov, 2007; Wu et al., 2007). Optionally multiple passes over the input are conducted until no tokens are left unattached (Attardi et al., 2007).

As an alternative to deterministic parsing, several parsers use probabilistic models and maintain a heap or beam of partial transition sequences in order to pick the most probable one at the end of the sentence

[6] They also flip the order of the labeler and the reranker.

[7] For diversity in parser ensembles, right-to-left parsers are also used.

(Duan et al., 2007; Johansson and Nugues, 2007b; Sagae and Tsujii, 2007; Titov and Henderson, 2007).

One system uses as part of their parsing pipeline a "neighbor-parser" that attaches adjacent words and a "root-parser" that identifies the root word(s) of a sentence (Wu et al., 2007). In the case of grammar-based parsers, a classifier is used to disambiguate in cases where the grammar leaves some ambiguity (Schneider et al., 2007; Watson and Briscoe, 2007)

5.2.3 Learning

Transition-based parsers either maintain a classifier that predicts the next transition or a global probabilistic model that scores a complete parse. To train these classifiers and probabilitistic models several approaches were used: SVMs (Duan et al., 2007; Hall et al., 2007a; Sagae and Tsujii, 2007), modified finite Newton SVMs (Wu et al., 2007), maximum entropy models (Sagae and Tsujii, 2007), multiclass averaged perceptron (Attardi et al., 2007) and maximum likelihood estimation (Watson and Briscoe, 2007).

In order to calculate a global score or probability for a transition sequence, two systems used a Markov chain approach (Duan et al., 2007; Sagae and Tsujii, 2007). Here probabilities from the output of a classifier are multiplied over the whole sequence of actions. This results in a locally normalized model. Two other entries used MIRA (Mannem, 2007) or online passive-aggressive learning (Johansson and Nugues, 2007b) to train a globally normalized model. Titov and Henderson (2007) used an incremental sigmoid Bayesian network to model the probability of a transition sequence and estimated model parameters using neural network learning.

5.3 Graph-Based Parsers

While transition-based parsers use training data to learn a process for deriving dependency graphs, graph-based parsers learn a model of what it means to be a good dependency graph given an input sentence. They define a scoring or probability function over the set of possible parses. At learning time they estimate parameters of this function; at parsing time they search for the graph that maximizes this function. These parsers mainly differ in the type and structure of the scoring function (model), the search algorithm that finds the best parse (inference), and the method to estimate the function's parameters (learning).

5.3.1 Models

The simplest type of model is based on a sum of local attachment scores, which themselves are calculated based on the dot product of a weight vector and a feature representation of the attachment. This type of scoring function is often referred to as a first-order model.[8] Several systems participating in this year's shared task used first-order models (Schiehlen and Spranger, 2007; Nguyen et al., 2007; Shimizu and Nakagawa, 2007; Hall et al., 2007b). Canisius and Tjong Kim Sang (2007) cast the same type of arc-based factorization as a weighted constraint satisfaction problem.

Carreras (2007) extends the first-order model to incorporate a sum over scores for pairs of adjacent arcs in the tree, yielding a second-order model. In contrast to previous work where this was constrained to sibling relations of the dependent (McDonald and Pereira, 2006), here head-grandchild relations can be taken into account.

In all of the above cases the scoring function is decomposed into functions that score local properties (arcs, pairs of adjacent arcs) of the graph. By contrast, the model of Nakagawa (2007) considers global properties of the graph that can take multiple arcs into account, such as multiple siblings and children of a node.

5.3.2 Inference

Searching for the highest scoring graph (usually a tree) in a model depends on the factorization chosen and whether we are looking for projective or non-projective trees. Maximum spanning tree algorithms can be used for finding the highest scoring non-projective tree in a first-order model (Hall et al., 2007b; Nguyen et al., 2007; Canisius and Tjong Kim Sang, 2007; Shimizu and Nakagawa, 2007), while Eisner's dynamic programming algorithm solves the problem for a first-order factorization in the projective case (Schiehlen and Spranger, 2007). Carreras (2007) employs his own extension of Eisner's algorithm for the case of projective trees and second-order models that include head-grandparent relations.

[8] It is also known as an edge-factored model.

The methods presented above are mostly efficient and always exact. However, for models that take global properties of the tree into account, they cannot be applied. Instead Nakagawa (2007) uses Gibbs sampling to obtain marginal probabilities of arcs being included in the tree using his global model and then applies a maximum spanning tree algorithm to maximize the sum of the logs of these marginals and return a valid cycle-free parse.

5.3.3 Learning

Most of the graph-based parsers were trained using an online inference-based method such as passive-aggressive learning (Nguyen et al., 2007; Schiehlen and Spranger, 2007), averaged perceptron (Carreras, 2007), or MIRA (Shimizu and Nakagawa, 2007), while some systems instead used methods based on maximum conditional likelihood (Nakagawa, 2007; Hall et al., 2007b).

5.4 Domain Adaptation

5.4.1 Feature-Based Approaches

One way of adapting a learner to a new domain without using any unlabeled data is to only include features that are expected to transfer well (Dredze et al., 2007). In structural correspondence learning a transformation from features in the source domain to features of the target domain is learnt (Shimizu and Nakagawa, 2007). The original source features along with their transformed versions are then used to train a discriminative parser.

5.4.2 Ensemble-Based Approaches

Dredze et al. (2007) trained a diverse set of parsers in order to improve cross-domain performance by incorporating their predictions as features for another classifier. Similarly, two parsers trained with different learners and search directions were used in the co-learning approach of Sagae and Tsujii (2007). Unlabeled target data was processed with both parsers. Sentences that both parsers agreed on were then added to the original training data. This combined data set served as training data for one of the original parsers to produce the final system. In a similar fashion, Watson and Briscoe (2007) used a variant of self-training to make use of the unlabeled target data.

5.4.3 Other Approaches

Attardi et al. (2007) learnt tree revision rules for the target domain by first parsing unlabeled target data using a strong parser; this data was then combined with labeled source data; a weak parser was applied to this new dataset; finally tree correction rules are collected based on the mistakes of the weak parser with respect to the gold data and the output of the strong parser.

Another technique used was to filter sentences of the out-of-domain corpus based on their similarity to the target domain, as predicted by a classifier (Dredze et al., 2007). Only if a sentence was judged similar to target domain sentences was it included in the training set.

Bick (2007) used a hybrid approach, where a data-driven parser trained on the labeled training data was given access to the output of a Constraint Grammar parser for English run on the same data. Finally, Schneider et al. (2007) learnt collocations and relational nouns from the unlabeled target data and used these in their parsing algorithm.

6 Analysis

Having discussed the major approaches taken in the two tracks of the shared task, we will now return to the test results. For the multilingual track, we compare results across data sets and across systems, and report results from a parser combination experiment involving all the participating systems (section 6.1). For the domain adaptation track, we sum up the most important findings from the test results (section 6.2).

6.1 Multilingual Track

6.1.1 Across Data Sets

The average LAS over all systems varies from 68.07 for Basque to 80.95 for English. Top scores vary from 76.31 for Greek to 89.61 for English. In general, there is a good correlation between the top scores and the average scores. For Greek, Italian, and Turkish, the top score is closer to the average score than the average distance, while for Czech, the distance is higher. The languages that produced the most stable results in terms of system ranks with respect to LAS are Hungarian and Italian. For UAS, Catalan also falls into this group. The language that

Setup	Arabic	Chinese	Czech	Turkish
2006 without punctuation	66.9	90.0	80.2	65.7
2007 without punctuation	75.5	84.9	80.0	71.6
2006 with punctuation	67.0	90.0	80.2	73.8
2007 with punctuation	76.5	84.7	80.2	79.8

Table 5: A comparison of the LAS top scores from 2006 and 2007. Official scoring conditions in boldface. For Turkish, scores with punctuation also include word-internal dependencies.

produced the most unstable results with respect to LAS is Turkish.

In comparison to last year's languages, the languages involved in the multilingual track this year can be more easily separated into three classes with respect to top scores:

- Low (76.31–76.94):
  Arabic, Basque, Greek
- Medium (79.19–80.21):
  Czech, Hungarian, Turkish
- High (84.40–89.61):
  Catalan, Chinese, English, Italian

It is interesting to see that the classes are more easily definable via language characteristics than via characteristics of the data sets. The split goes across training set size, original data format (constituent vs. dependency), sentence length, percentage of unknown words, number of dependency labels, and ratio of (C)POSTAGS and dependency labels. The class with the highest top scores contains languages with a rather impoverished morphology. Medium scores are reached by the two agglutinative languages, Hungarian and Turkish, as well as by Czech. The most difficult languages are those that combine a relatively free word order with a high degree of inflection. Based on these characteristics, one would expect to find Czech in the last class. However, the Czech training set is four times the size of the training set for Arabic, which is the language with the largest training set of the difficult languages.

However, it would be wrong to assume that training set size alone is the deciding factor. A closer look at table 1 shows that while Basque and Greek in fact have small training data sets, so do Turkish and Italian. Another factor that may be associated with the above classification is the percentage of new words (PNW) in the test set. Thus, the expectation would be that the highly inflecting languages have a high PNW while the languages with little morphology have a low PNW. But again, there is no direct correspondence. Arabic, Basque, Catalan, English, and Greek agree with this assumption: Catalan and English have the smallest PNW, and Arabic, Basque, and Greek have a high PNW. But the PNW for Italian is higher than for Arabic and Greek, and this is also true for the percentage of new lemmas. Additionally, the highest PNW can be found in Hungarian and Turkish, which reach higher scores than Arabic, Basque, and Greek. These considerations suggest that highly inflected languages with (relatively) free word order need more training data, a hypothesis that will have to be investigated further.

There are four languages which were included in the shared tasks on multilingual dependency parsing both at CoNLL 2006 and at CoNLL 2007: Arabic, Chinese, Czech, and Turkish. For all four languages, the same treebanks were used, which allows a comparison of the results. However, in some cases the size of the training set changed, and at least one treebank, Turkish, underwent a thorough correction phase. Table 5 shows the top scores for LAS. Since the official scores excluded punctuation in 2006 but includes it in 2007, we give results both with and without punctuation for both years.

For Arabic and Turkish, we see a great improvement of approximately 9 and 6 percentage points. For Arabic, the number of tokens in the training set doubled, and the morphological annotation was made more informative. The combined effect of these changes can probably account for the substantial improvement in parsing accuracy. For Turkish, the training set grew in size as well, although only by 600 sentences, but part of the improvement for Turkish may also be due to continuing efforts in error correction and consistency checking. We see that the choice to include punctuation or not makes a large difference for the Turkish scores, since non-final IGs of a word are counted as punctuation (because they have the underscore character as their FORM value), which means that word-internal dependency links are included if punctuation is included.[9] However, regardless of whether we compare scores with or without punctuation, we see a genuine improvement of approximately 6 percentage points.

For Chinese, the same training set was used. Therefore, the drop from last year's top score to this year's is surprising. However, last year's top scoring system for Chinese (Riedel et al., 2006), which did not participate this year, had a score that was more than 3 percentage points higher than the second best system for Chinese. Thus, if we compare this year's results to the second best system, the difference is approximately 2 percentage points. This final difference may be attributed to the properties of the test sets. While last year's test set was taken from the treebank, this year's test set contains texts from other sources. The selection of the textual basis also significantly changed average sentence length: The Chinese training set has an average sentence length of 5.9. Last year's test set also had an average sentence length of 5.9. However, this year, the average sentence length is 7.5 tokens, which is a significant increase. Longer sentences are typically harder to parse due to the increased likelihood of ambiguous constructions.

Finally, we note that the performance for Czech is almost exactly the same as last year, despite the fact that the size of the training set has been reduced to approximately one third of last year's training set. It is likely that this in fact represents a relative improvement compared to last year's results.

6.1.2 Across Systems

The LAS over all languages ranges from 80.32 to 54.55. The comparison of the system ranks averaged over all languages with the ranks for single languages show considerably more variation than last year's systems. Buchholz and Marsi (2006) report that "[f]or most parsers, their ranking differs at most a few places from their overall ranking". This year, for all of the ten best performing systems with respect to LAS, there is at least one language for which their rank is at least 5 places different from their overall rank. The most extreme case is the top performing Nilsson system (Hall et al., 2007a), which reached rank 1 for five languages and rank 2 for two more languages. Their only outlier is for Chinese, where the system occupies rank 14, with a LAS approximately 9 percentage points below the top scoring system for Chinese (Sagae and Tsujii, 2007). However, Hall et al. (2007a) point out that the official results for Chinese contained a bug, and the true performance of their system was actually much higher. The greatest improvement of a system with respect to its average rank occurs for English, for which the system by Nguyen et al. (2007) improved from the average rank 15 to rank 6. Two more outliers can be observed in the system of Johansson and Nugues (2007b), which improves from its average rank 12 to rank 4 for Basque and Turkish. The authors attribute this high performance to their parser's good performance on small training sets. However, this hypothesis is contradicted by their results for Greek and Italian, the other two languages with small training sets. For these two languages, the system's rank is very close to its average rank.

6.1.3 An Experiment in System Combination

Having the outputs of many diverse dependency parsers for standard data sets opens up the interesting possibility of parser combination. To combine the outputs of each parser we used the method of Sagae and Lavie (2006). This technique assigns to each possible labeled dependency a weight that is equal to the number of systems that included the dependency in their output. This can be viewed as an arc-based voting scheme. Using these weights it is possible to search the space of possible dependency trees using directed maximum spanning tree algorithms (McDonald et al., 2005). The maximum spanning tree in this case is equal to the tree that on average contains the labeled dependencies that most systems voted for. It is worth noting that variants of this scheme were used in two of the participating

[9]The decision to include word-internal dependencies in this way can be debated on the grounds that they can be parsed deterministically. On the other hand, they typically correspond to regular dependencies captured by function words in other languages, which are often easy to parse as well. It is therefore unclear whether scores are more inflated by including word-internal dependencies or deflated by excluding them.

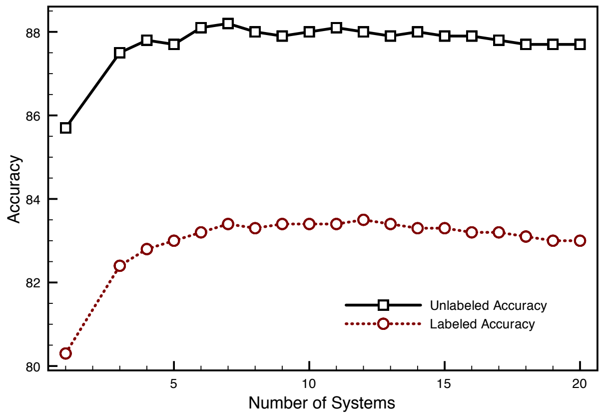

Figure 1: System Combination

systems, the Nilsson system (Hall et al., 2007a) and the system of Sagae and Tsujii (2007).

Figure 1 plots the labeled and unlabeled accuracies when combining an increasing number of systems. The data used in the plot was the output of all competing systems for every language in the multilingual track. The plot was constructed by sorting the systems based on their average labeled accuracy scores over all languages, and then incrementally adding each system in descending order.[10] We can see that both labeled and unlabeled accuracy are significantly increased, even when just the top three systems are included. Accuracy begins to degrade gracefully after about ten different parsers have been added. Furthermore, the accuracy never falls below the performance of the top three systems.

6.2 Domain Adaptation Track

For this task, the results are rather surprising. A look at the LAS and UAS for the chemical research abstracts shows that there are four closed systems that outperform the best scoring open system. The best system (Sagae and Tsujii, 2007) reaches an LAS of 81.06 (in comparison to their LAS of 89.01 for the English data set in the multilingual track). Considering that approximately one third of the words of the chemical test set are new, the results are noteworthy.

The next surprise is to be found in the relatively low UAS for the CHILDES data. At a first glance, this data set has all the characteristics of an easy set; the average sentence is short (12.9 words), and the percentage of new words is also small (6.10%). Despite these characteristics, the top UAS reaches 62.49 and is thus more than 10 percentage points below the top UAS for the chemical data set. One major reason for this is that auxiliary and main verb dependencies are annotated differently in the CHILDES data than in the WSJ training set. As a result of this discrepancy, participants were not required to submit results for the CHILDES data. The best performing system on the CHILDES corpus is an open system (Bick, 2007), but the distance to the top closed system is approximately 1 percentage point. In this domain, it seems more feasible to use general language resources than for the chemical domain. However, the results prove that the extra effort may be unnecessary.

7 Conclusion

Two years of dependency parsing in the CoNLL shared task has brought an enormous boost to the development of dependency parsers for multiple languages (and to some extent for multiple domains). But even though nineteen languages have been covered by almost as many different parsing and learning approaches, we still have only vague ideas about the strengths and weaknesses of different methods for languages with different typological characteristics. Increasing our knowledge of the multi-causal relationship between language structure, annotation scheme, and parsing and learning methods probably remains the most important direction for future research in this area. The outputs of all systems for all data sets from the two shared tasks are freely available for research and constitute a potential gold mine for comparative error analysis across languages and systems.

For domain adaptation we have barely scratched the surface so far. But overcoming the bottleneck of limited annotated resources for specialized domains will be as important for the deployment of human language technology as being able to handle multiple languages in the future. One result from the domain adaptation track that may seem surprising at first is the fact that closed class systems outperformed open class systems on the chemical abstracts. However, it seems that the major problem in

[10]The reason that there is no data point for two parsers is that the simple voting scheme adopted only makes sense with at least three parsers voting.

adapting pre-existing parsers to the new domain was not the domain as such but the mapping from the native output of the parser to the kind of annotation provided in the shared task data sets. Thus, finding ways of reusing already invested development efforts by adapting the outputs of existing systems to new requirements, without substantial loss in accuracy, seems to be another line of research that may be worth pursuing.

Acknowledgments

First and foremost, we want to thank all the people and organizations that generously provided us with treebank data and helped us prepare the data sets and without whom the shared task would have been literally impossible: Otakar Smrz, Charles University, and the LDC (Arabic); Maxux Aranzabe, Kepa Bengoetxea, Larraitz Uria, Koldo Gojenola, and the University of the Basque Country (Basque); Ma. Antònia Martí Antonín, Lluís Màrquez, Manuel Bertran, Mariona Taulé, Difda Monterde, Eli Comelles, and CLiC-UB (Catalan); Shih-Min Li, Keh-Jiann Chen, Yu-Ming Hsieh, and Academia Sinica (Chinese); Jan Hajič, Zdenek Zabokrtsky, Charles University, and the LDC (Czech); Brian MacWhinney, Eric Davis, the CHILDES project, the Penn BioIE project, and the LDC (English); Prokopis Prokopidis and ILSP (Greek); Csirik János and Zoltán Alexin (Hungarian); Giuseppe Attardi, Simonetta Montemagni, Maria Simi, Isidoro Barraco, Patrizia Topi, Kiril Ribarov, Alessandro Lenci, Nicoletta Calzolari, ILC, and ELRA (Italian); Gülşen Eryiğit, Kemal Oflazer, and Ruket Çakıcı (Turkish).

Secondly, we want to thank the organizers of last year's shared task, Sabine Buchholz, Amit Dubey, Erwin Marsi, and Yuval Krymolowski, who solved all the really hard problems for us and answered all our questions, as well as our colleagues who helped review papers: Jason Baldridge, Sabine Buchholz, James Clarke, Gülşen Eryiğit, Kilian Evang, Julia Hockenmaier, Yuval Krymolowski, Erwin Marsi, Beáta Megyesi, Yannick Versley, and Alexander Yeh. Special thanks to Bertjan Busser and Erwin Marsi for help with the CoNLL shared task website and many other things, and to Richard Johansson for letting us use his conversion tool for English.

Thirdly, we want to thank the program chairs for EMNLP-CoNLL 2007, Jason Eisner and Taku Kudo, the publications chair, Eric Ringger, the SIGNLL officers, Antal van den Bosch, Hwee Tou Ng, and Erik Tjong Kim Sang, and members of the LDC staff, Tony Castelletto and Ilya Ahtaridis, for great cooperation and support.

Finally, we want to thank the following people, who in different ways assisted us in the organization of the CoNLL 2007 shared task: Giuseppe Attardi, Eckhard Bick, Matthias Buch-Kromann, Xavier Carreras, Tomaz Erjavec, Svetoslav Marinov, Wolfgang Menzel, Xue Nianwen, Gertjan van Noord, Petya Osenova, Florian Schiel, Kiril Simov, Zdenka Uresova, and Heike Zinsmeister.

References

A. Abeillé, editor. 2003. *Treebanks: Building and Using Parsed Corpora*. Kluwer.

I. Aduriz, M. J. Aranzabe, J. M. Arriola, A. Atutxa, A. Diaz de Ilarraza, A. Garmendia, and M. Oronoz. 2003. Construction of a Basque dependency treebank. In *Proc. of the 2nd Workshop on Treebanks and Linguistic Theories (TLT)*.

G. Attardi, F. Dell'Orletta, M. Simi, A. Chanev, and M. Ciaramita. 2007. Multilingual dependency parsing and domain adaptation using desr. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

E. Bick. 2007. Hybrid ways to improve domain independence in an ML dependency parser. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

J. Blitzer, R. McDonald, and F. Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*.

A. Böhmová, J. Hajič, E. Hajičová, and B. Hladká. 2003. The PDT: a 3-level annotation scenario. In Abeillé (2003), chapter 7, pages 103–127.

R. Brown. 1973. *A First Language: The Early Stages*. Harvard University Press.

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

S. Canisius and E. Tjong Kim Sang. 2007. A constraint satisfaction approach to dependency parsing. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

X. Carreras. 2007. Experiments with a high-order projective dependency parser. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

C. Chelba and A. Acero. 2004. Adaptation of maximum entropy capitalizer: Little data can help a lot. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*.

K. Chen, C. Luo, M. Chang, F. Chen, C. Chen, C. Huang, and Z. Gao. 2003. Sinica treebank: Design criteria, representational issues and implementation. In Abeillé (2003), chapter 13, pages 231–248.

W. Chen, Y. Zhang, and H. Isahara. 2007. A two-stage parser for multilingual dependency parsing. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

M. Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proc. of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*.

M. A. Covington. 2001. A fundamental algorithm for dependency parsing. In *Proc. of the 39th Annual ACM Southeast Conf.*

D. Csendes, J. Csirik, T. Gyimóthy, and A. Kocsor. 2005. *The Szeged Treebank*. Springer.

H. Daumé and D. Marcu. 2006. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126.

M. Dredze, J. Blitzer, P. P. Talukdar, K. Ganchev, J. Graca, and F. Pereira. 2007. Frustratingly hard domain adaptation for dependency parsing. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

X. Duan, J. Zhao, and B. Xu. 2007. Probabilistic parsing action models for multi-lingual dependency parsing. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

G. Eryiğit. 2007. ITU validation set for Metu-Sabancı Turkish Treebank. URL: http://www3.itu.edu.tr/∼gulsenc/papers/validationset.pdf.

R. Florian, H. Hassan, A. Ittycheriah, H. Jing, N. Kambhatla, A. Luo, N. Nicolov, and S. Roukos. 2004. A statisical model for multilingual entity detection and tracking. In *Proc. of the Human Language Technology Conf. and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*.

D. Gildea. 2001. Corpus variation and parser performance. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP)*.

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 Intern. Conf. on Arabic Language Resources and Tools*.

J. Hall, J. Nilsson, J. Nivre, G. Eryiğit, B. Megyesi, M. Nilsson, and M. Saers. 2007a. Single malt or blended? A study in multilingual parser optimization. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

K. Hall, J. Havelka, and D. Smith. 2007b. Log-linear models of non-projective trees, k-best MST parsing and tree-ranking. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

R. Johansson and P. Nugues. 2007a. Extended constituent-to-dependency conversion for English. In *Proc. of the 16th Nordic Conf. on Computational Linguistics (NODALIDA)*.

R. Johansson and P. Nugues. 2007b. Incremental dependency parsing using online learning. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

T. Kudo and Y. Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proc. of the Sixth Conf. on Computational Language Learning (CoNLL)*.

S. Kulick, A. Bies, M. Liberman, M. Mandel, R. McDonald, M. Palmer, A. Schein, and L. Ungar. 2004. Integrated annotation for biomedical information extraction. In *Proc. of the Human Language Technology Conf. and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*.

B. MacWhinney. 2000. *The CHILDES Project: Tools for Analyzing Talk*. Lawrence Erlbaum.

P. R. Mannem. 2007. Online learning for deterministic dependency parsing. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

M. Marcus, B. Santorini, and M. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

S. Marinov. 2007. Covington variations. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

M. A. Martí, M. Taulé, L. Màrquez, and M. Bertran. 2007. CESS-ECE: A multilingual and multilevel annotated corpus. Available for download from: http://www.lsi.upc.edu/∼mbertran/cess-ece/.

D. McClosky, E. Charniak, and M. Johnson. 2006. Reranking and self-training for parser adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.

R. McDonald and J. Nivre. 2007. Characterizing the errors of data-driven dependency parsing models. In *Proc. of the Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

R. McDonald and F. Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proc. of the 11th Conf. of the European Chapter of the Association for Computational Linguistics (EACL)*.

R. McDonald, F. Pereira, K. Ribarov, and J. Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proc. of the Human Language Technology Conf. and the Conf. on Empirical Methods in Natural Language Processing (HLT/EMNLP)*.

S. Montemagni, F. Barsotti, M. Battista, N. Calzolari, O. Corazzari, A. Lenci, A. Zampolli, F. Fanciulli, M. Massetani, R. Raffaelli, R. Basili, M. T. Pazienza, D. Saracino, F. Zanzotto, N. Nana, F. Pianesi, and R. Delmonte. 2003. Building the Italian Syntactic-Semantic Treebank. In Abeillé (2003), chapter 11, pages 189–210.

T. Nakagawa. 2007. Multilingual dependency parsing using Gibbs sampling. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

L.-M. Nguyen, T.-P. Nguyen, and A. Shimazu. 2007. A multilingual dependency analysis system using online passive-aggressive learning. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryiğit, S. Kübler, S. Marinov, and E. Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13:95–135.

K. Oflazer, B. Say, D. Zeynep Hakkani-Tür, and G. Tür. 2003. Building a Turkish treebank. In Abeillé (2003), chapter 15, pages 261–277.

P. Prokopidis, E. Desypri, M. Koutsombogera, H. Papageorgiou, and S. Piperidis. 2005. Theoretical and practical issues in the construction of a Greek dependency treebank. In *Proc. of the 4th Workshop on Treebanks and Linguistic Theories (TLT)*.

S. Riedel, R. Çakıcı, and I. Meza-Ruiz. 2006. Multilingual dependency parsing with incremental integer linear programming. In *Proc. of the Tenth Conf. on Computational Natural Language Learning (CoNLL)*.

B. Roark and M. Bacchiani. 2003. Supervised and unsupervised PCFG adaptation to novel domains. In *Proc. of the Human Language Technology Conf. and the Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*.

K. Sagae and A. Lavie. 2006. Parser combination by reparsing. In *Proc. of the Human Language Technology Conf. of the North American Chapter of the Association of Computational Linguistics (HLT/NAACL)*.

K. Sagae and J. Tsujii. 2007. Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

M. Schiehlen and Kristina Spranger. 2007. Global learning of labelled dependency trees. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

G. Schneider, K. Kaljurand, F. Rinaldi, and T. Kuhn. 2007. Pro3Gres parser in the CoNLL domain adaptation shared task. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

N. Shimizu and H. Nakagawa. 2007. Structural correspondence learning for dependency parsing. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

I. Titov and J. Henderson. 2006. Porting statistical parsers with data-defined kernels. In *Proc. of the Tenth Conf. on Computational Natural Language Learning (CoNLL)*.

I. Titov and J. Henderson. 2007. Fast and robust multilingual dependency parsing with a generative latent variable model. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

R. Watson and T. Briscoe. 2007. Adapting the RASP system for the CoNLL07 domain-adaptation task. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

Y.-C. Wu, J.-C. Yang, and Y.-S. Lee. 2007. Multilingual deterministic dependency parsing framework using modified finite Newton method support vector machines. In *Proc. of the CoNLL 2007 Shared Task. EMNLP-CoNLL*.

H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proc. 8th International Workshop on Parsing Technologies (IWPT)*.