Dutch Dependency Parser Performance Across Domains

Barbara Plank and Gertjan van Noord

Computational Linguistics, Faculty of Arts, University of Groningen

Abstract

In the past decade several natural language parsing systems have emerged, which use different methods and formalisms. For instance, systems that employ a hand-crafted grammar with a statistical disambiguation component versus purely statistical data-driven systems. What they have in common is the lack of portability to new domains: their performance might decrease substantially as the distance between test and training domain increases. Yet, to which degree do they suffer from this problem, i.e. which kind of parsing system is more affected by domain shifts? To address this question, we evaluate the performance variation of two kinds of dependency parsing systems for Dutch (grammar-driven versus data-driven) across several domains. We examine (1) how parser performance correlates to simple statistical properties of the text and (2) how sensitive a given system is to the text domain. This will give us an estimate of which kind of system is more affected by domain shifts, and thus more in need for domain adaptation techniques. To this end, we extend the statistical measures used by Zhang and Wang (2009a) for English and propose a new simple measure to quantify domain sensitivity.

1 Introduction

Most modern Natural Language Processing (NLP) systems are subject to the lack of portability to new domains: there is a substantial drop in their performance when the system gets input from another text domain (Gildea 2001). This is the problem of *domain adaptation*. Although this problem exists ever since the emergence of supervised Machine Learning, it has started to get attention only in recent years.

Studies on *supervised domain adaptation* (where there are limited amounts of annotated resources in the new domain) have shown that straightforward baselines (e.g. models based on source only, target only, or the union of the data) achieve a relatively high performance level and are "surprisingly difficult to beat" (Daumé III 2007). In contrast, *semi-supervised adaptation* (i.e. no annotated resources in the new domain) is a much more realistic situation but is also considerably more difficult. Current studies on semi-supervised approaches show very mixed results. Dredze et al. (2007) report on "frustrating" results on the CoNLL 2007 semi-supervised adaptation task for dependency parsing, i.e. "no team was able to improve target domain performance substantially over a state-of-the-art baseline". On the other hand, there have been positive results as well. For instance, McClosky et al. (2006) improved a statistical parser by self-training. Structural Correspondence Learning (Blitzer et al. 2006) was effective for PoS tagging and Sentiment

Proceedings of the 20th Meeting of Computational Linguistics in the Netherlands
Edited by: Eline Westerhout, Thomas Markus, and Paola Monachesi.

Classification (Blitzer et al. 2006, Blitzer et al. 2007), while only modest gains were obtained for structured output tasks like parsing.

For parsing, most previous work on domain adaptation has focused on *data-driven* systems (Gildea 2001, McClosky et al. 2006, Dredze et al. 2007), i.e. systems employing (constituent or dependency based) treebank grammars. Only few studies examined the adaptation of *grammar-based* systems (Hara et al. 2005, Plank and van Noord 2008), i.e. systems employing a hand-crafted grammar with a statistical disambiguation component. This may be motivated by the fact that potential gains for this task are inherently bound by the underlying grammar. Yet, domain adaptation poses a challenge for both kinds of parsing systems. But to what extent do these different kinds of parsing systems suffer from the problem? To address this question, we examine two particular issues:[1]

(Q.1) How does parser performance for Dutch correlate to simple statistical measures of the text?

(Q.2) How sensitive is a given system to the text domain, i.e. which parsing system (hand-crafted versus purely statistical) is more affected by domain shifts, and thus more in need for adaptation techniques?

The remainder of this paper is structured as follows. Section 2 provides an overview of related work. Section 3 and 4 introduce the different parsing systems, datasets and the experimental setup. Next, Q.1 is addressed in Section 5. Section 6 focuses on Q.2 and proposes a new simple measure to quantify domain sensitivity. In Section 7, conclusions are drawn and directions for future work are presented.

2 Related Work

For the statistical measures of the text and their correlation to parsing accuracy (Q.1) we start here from work by Zhang and Wang (2009a), who examined several state-of-the-art parsing models for English (WSJ and Brown). They show that different parsing models (constituent, dependency and deep-grammar based system) correlate on different levels to the three statistical measures examined (average sentence length, unknown word ratio and unknown part-of-speech trigram ratio). Their work directly inspired us. A related study is Ravi et al. (2008), who built a regression model to predict parser accuracy for English constituent parsing.

Regarding our second question (Q.2), to the best of our knowledge, no study has yet addressed this issue. Most previous work has focused on a single parsing system in isolation (Gildea 2001, Hara et al. 2005, McClosky et al. 2006). Recently, there is an observable trend towards combining different parsing systems to exploit complementary strengths. For instance, Nivre and McDonald (2008) combine two data-driven systems to improve dependency accuracy. Similarly, two studies successfully combined grammar-based and data-driven systems: Sagae

[1] Preliminary results of these research questions have been reported in Plank (2010) and Plank and van Noord (2010).

et al. (2007) incorporate data-driven dependencies as soft-constraint in a HPSG-based system for parsing the Wallstreet Journal. In the same spirit (but the other direction), Zhang and Wang (2009b) use a deep-grammar based backbone to improve data-driven parsing accuracy. They incorporate features from the grammar-based backbone into the data-driven system to achieve better generalization across domains. However, one issue remains open: which kind of system (hand-crafted versus purely statistical) is more affected by the domain, and thus more sensitive to domain shifts? We present an empirical evaluation of different parsing systems for Dutch, and propose a new simple measure to quantify domain sensitivity.

3 Parsing Systems

The parsing systems used in this study are: a grammar-based system coupled with a statistical disambiguation system (Alpino) and two data-driven systems (MST and Malt), described in the sequel.

(1) *Alpino* (van Noord 2006) is a deep-grammar based parser for Dutch that produces dependency structures as output. The system consists of approximately 800 grammar rules in the tradition of HPSG, and a large hand-crafted lexicon, that together with a left-corner parser constitutes the parser component. For words that are not in the lexicon, the system applies a large variety of unknown word heuristics (van Noord 2006), which among others attempt to deal with number-like expressions, compounds and proper names. The second stage of Alpino is a statistical disambiguation component based on Maximum Entropy. Thus, training the parser requires estimating parameters for the disambiguation component.

(2) *MST Parser* (McDonald et al. 2005) is a data-driven graph-based dependency parser. The system couples a minimum spanning tree search procedure with a separate second stage classifier to label the dependency edges.

(3) *MALT Parser* (Nivre et al. 2007) is a data-driven transition-based dependency parser. Malt parser uses SVMs to learn a classifier that predicts the next parsing action. Training instances represent parser configurations and the label to predict determines the next parser action.

Both data-driven parsers (MST and Malt) are thus not specific for the Dutch language, however, they can be trained on a variety of languages given that the training corpus complies with the column-based format introduced in the 2006 CoNLL shared task (Buchholz and Marsi 2006). Additionally, both parsers implement projective and non-projective parsing algorithms, where the latter will be used in our experiments on the relatively free word order language Dutch. Despite that, we train the data-driven parsers using their default settings (e.g. first order features for MST, SVM with polynomial kernel for Malt).

4 Datasets and Experimental Setup

The source domain on which all parsers are trained is cdb, the newspaper part of the Alpino Treebank (van Noord 2006). For our cross-domain evaluation, we consider Wikipedia and the Dutch Parallel Corpus (DPC). All are described next.

Source Cdb is a collection of text fragments from 6 Dutch newspapers, which has been annotated according to the guidelines of CGN (Oostdijk 2000) and stored in XML format. It consists of 140,000 words (7,136 sentences; average sentence length of 19.7 words). It is the standard treebank used to train the disambiguation component of the Alpino parser. Note that cdb is a subset of the training corpus used in the CoNLL 2006 shared task (Buchholz and Marsi 2006). The CoNLL data additionally contained a mix of non-newspaper text (namely, a large amount of questions from CLEF, roughly 4k questions, and around 1.5k hand-crafted sentences used during the development of the grammar), which we exclude here on purpose to keep a clean baseline.

Target The Wikipedia and DPC subpart of the LASSY corpus[2] constitute our target domains. These corpora contain several subdomains, e.g. sports, locations, science, communication (in total 10 Wikipedia and 13 DPC subdomains). A detailed overview of the corpora is given in Table 8.1. Note that both consist of hand-corrected data labeled by Alpino. This might introduce a slight bias towards Alpino, however it has the advantage that all domains employ the same annotation scheme. This avoids the problem of having differences in annotation guidelines, which was the major source of error in the CoNLL 2007 shared task on domain adaptation (Dredze et al. 2007).

CoNLL2006 This is the test file for Dutch that has been used in the CoNLL 2006 shared task on multi-lingual dependency parsing. The file consists of 386 sentences from an institutional brochure (about 'Jeugdgezondheidszorg'/youth healthcare) with an average sentence length of 15.2 words. We will use this file to check our data-driven models against state-of-the-art performance (Section 5.1).

Alpino to CoNLL format In order to train the MST and Malt parser and evaluate it on the various Wikipedia and DPC articles, we needed to convert the Alpino Treebank format into the tabular CoNLL format. To this end, we adapted the treebank conversion software developed by Erwin Marsi for the CoNLL 2006 shared task on multi-lingual dependency parsing. Instead of using the PoS tagger and tagset used in the CoNLL shared task (to which we did not have access at the time of these experiments), we replaced the PoS tags with more fine-grained tags obtained by parsing the data with the Alpino parser.[3] At testing time, the data-driven parsers are given the PoS tagged data as input, while Alpino uses plain sentences.

Evaluation In all experiments, unless otherwise specified, performance is measured as Labeled Attachment Score (LAS), the percentage of tokens with the correct dependency edge and label. To compute LAS, we use the CoNLL 2007 eval-

[2]LASSY (Large Scale Syntactic Annotation of written Dutch), ongoing project. Corpus version 17905, obtained from `http://www.let.rug.nl/vannoord/Lassy/corpus/`

[3]As will be discussed later (Section 5.1, cf. Table 8.2), using Alpino tags actually improved the performance of the data-driven parsers significantly. We could perform this check as we recently got access to the tagger and tagset used in the CoNLL shared task (Mbt with wotan tagset; thanks to Erwin Marsi).

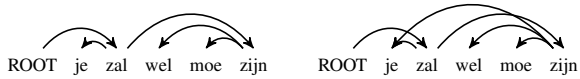
Wikipedia	**Wikipedia articles (excerpt)**	**#articles**	**#sentences**	**#words**	**ASL**
LOC (location)	België, Brussel (stad)	31	2190	25259	11.5
KUN (arts)	School van Tervuren	11	998	17073	17.1
POL (politics)	Belgische verkiezingen 2003	16	983	15107	15.4
SPO (sports)	Spa-Francorchamps, Kim Clijsters	9	877	9713	11.1
HIS (history)	Geschiedenis van België	3	468	8396	17.9
BUS (business)	Algemeen Belgisch Vakverbond	9	405	4440	11.0
NOB (nobility)	Albert II van België	6	277	4179	15.1
COM (comics)	Suske en Wiske	3	380	4000	10.5
MUS (music)	Sandra Kim, Urbanus (artiest)	3	89	1296	14.6
HOL (holidays)	Feest Vlaamse Gemeenschap	4	43	524	12.2
Total		**95**	**6710**	**89987**	**13.4**

DPC	**Description/Example articles**	**#articles**	**#sentences**	**#words**	**ASL**
Science	medicine (Zyprexa); Oceanography	69	3159	60787	19.2
Institutions	politics (Toespraak MP Kok)	21	1777	28646	16.1
Communication	ICT/Inet (DNS registreren)	29	1524	26640	17.5
Welfare state	pensions (Informatie over de AOW)	22	1130	20198	17.9
Culture	background articles (darwinisme)	11	791	16237	20.5
Economy	business texts (Inflatie op maat)	9	794	14722	18.5
Education	school (onderwijs in vlaanderen)	2	733	11980	16.3
Home affairs	presentation (Brussel, je hoofdstad)	1	540	9340	17.3
Foreign affairs	EU (Toespraak Cox EU raad)	7	372	9007	24.2
Environment	threat/nature(Koude oorlog noord-pool)	6	419	8534	20.4
Finance	banks (Opleiding private bankiers)	6	275	6127	22.3
Leisure	various (seks- en drugsschandaal)	2	140	2843	20.3
Consumption	toys (speelgoed uit China)	1	58	1310	22.6
Total		**186**	**11712**	**216371**	**18.5**

Table 8.1: Overview Wikipedia and DPC corpus. ASL = average sentence length.

uation script[4] with punctuation tokens excluded from scoring (as was the default setting in CoNLL 2006). We thus evaluate all parsers using the same evaluation metric. Note that the standard metric for Alpino is a variant of LAS, which allows for a discrepancy between expected and returned dependencies. Such a discrepancy can occur, for instance, because the syntactic annotation of Alpino allows words to be dependent on more than a single head (van Noord 2006). However, such 'secondary edges' are ignored in the CoNLL format; just a single head per token is allowed. The following example illustrates this:

CoNLL: *Alpino:*



Furthermore, there is another simplification. As the Dutch tagger used in the CoNLL 2006 shared task did not have the concept of multiwords, the organizers chose to treat them as a single token (Buchholz and Marsi 2006). We here follow the CoNLL 2006 task setup. Thus, to evaluate the Alpino output we convert it to CoNLL format using the same software.

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

5 Parser Performance and Simple Measures of the Text

We follow Zhang and Wang (2009a) and look at this stage at simple characteristics of the dataset without looking at syntactic annotation. We are interested in their correlation to parsing performance for Dutch.

Statistical measures We depart from the measures of Zhang and Wang (2009a) and add a perplexity measure estimated from a word-trigram Language Model. Thus the statistical measures used are:

Average Sentence Length (ASL) measures the average sentence length. Intuitively, longer sentences should be more difficult to parse than shorter ones.

Simple Unknown Word Rate (sUWR) calculates how many words (tokens) in the dataset have not been observed before, i.e. are not in the cdb corpus. For the Alpino parser, we use the percentage of words that are not in the lexicon (*aUWR, Alpino Unknown Word Rate*).

Unknown PoS Trigram Ration (UPTR) calculates the number of unknown PoS trigrams with respect to the original cdb training data.

Perplexity is the perplexity score assigned by a word-trigram language model estimated from the original cdb training data using the SRILM toolkit[5]. This feature, also used by Ravi et al. (2008), is intended as a refinement of UWR.

5.1 Empirical Results

Sanity checks First of all, we performed several sanity checks. We trained the MST parser on the entire original CoNLL training data as well as the cdb subpart only, and evaluated it on the original CoNLL test data. As shown in Table 8.2 (row 1-2) the accuracies of both models falls slightly below state-of-the-art performance (row 5), most probably due to the fact that we used standard parsing settings (e.g. no second-order features for MST). More importantly, there was basically no difference in performance when trained on the entire data or cdb only.

Model	**LAS**	**UAS**
MST (original CoNLL)	78.35	82.89
MST (original CoNLL, cdb subpart)	78.37	82.71
MST (cdb retagged with Alpino)	82.14	85.51
Malt (cdb retagged with Alpino)	80.64	82.66
MST (Nivre and McDonald 2008)	79.19	83.6
Malt (Nivre and McDonald 2008)	78.59	n/a
MST (cdb retagged with Mbt)	78.73	82.66

Table 8.2: Performance of the data-driven parsers versus state-of-the-art performance (McDonald et al. 2005; Nivre & McDonald, 2008) on the CoNLL 2006 test set (in Labeled/Unlabeled Attachment Score).

[5] http://www-speech.sri.com/projects/srilm/

We then trained the MST and Malt parser on the cdb corpus converted into the retagged CoNLL format, and tested on CoNLL 2006 test data (also retagged with Alpino). As seen in Table 8.2 (row 3 to 6), using Alpino tags improves the performance level significantly ($p < 0.002$, Approximate Randomization Test with 1000 iterations). This increase in performance can be attributed to two sources: (a) improvements in the Alpino treebank itself over the course of the years, and (b) the more fine-grained PoS tagset obtained by parsing the data with the deep grammar. To examine the contribution of each source, we trained an additional MST model on the cdb data but tagged with the same tagger as in the CoNLL shared task (Mbt, cf. Table 8.2 last row): the results show that the major source of improvement actually comes from using the more fine-grained Alpino tags ($78.73 \rightarrow 82.14 = +3.41$ LAS), rather than the changes in the treebank ($78.37 \rightarrow 78.73 = +0.36$ LAS). Despite the rather limited training data and use of standard training settings, we are in line with (and actually above) current results of data-driven parsing for Dutch.

We now turn to the various statistical measures. The parsers were all evaluated on the 95 Wikipedia articles.

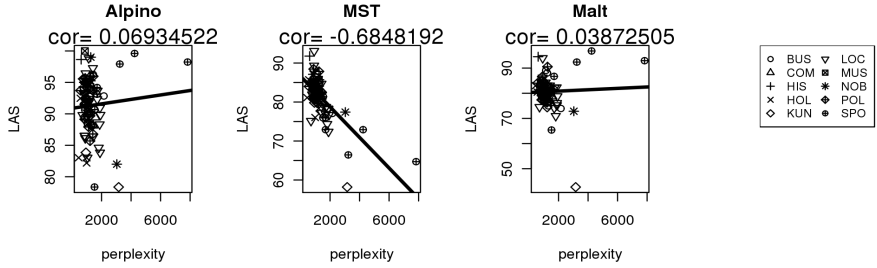


Figure 1: Pre-results (on 95 Wiki articles): Parser performance against sentence perplexity, including correlation coefficient – 3 peculiar sports articles fall out (crossed dots).

Pre-result While plotting the correlation between parser performance and statistical measure, three datasets immediately caught our eyes (the crossed dots; cf. Figure 1; we here only plot the perplexity graphs due to space reasons). These are three sports (SPO) articles about bike races. By inspecting them we notice that they contain a long list of winners from the various race years (on average 86% of the articles constitute this 'winner list'). Thus, despite the average short sentence length (6.03 words per sentence; in contrast to an average sentence length on Wikipedia of 13.4 words), the parsers exhibit very different performance levels on these datasets. Alpino, which includes various unknown word heuristics and a named entity tagger, is rather robust against the very high unknown word rate and reaches a very high accuracy level on these datasets. The Malt parser also reaches a high performance level on these special datasets. In contrast, the MST parser is

more influenced by unknown words, and its performance on these articles drops to its lowest level. These three sports articles thus form 'outliers' and we exclude them from the following experiment.

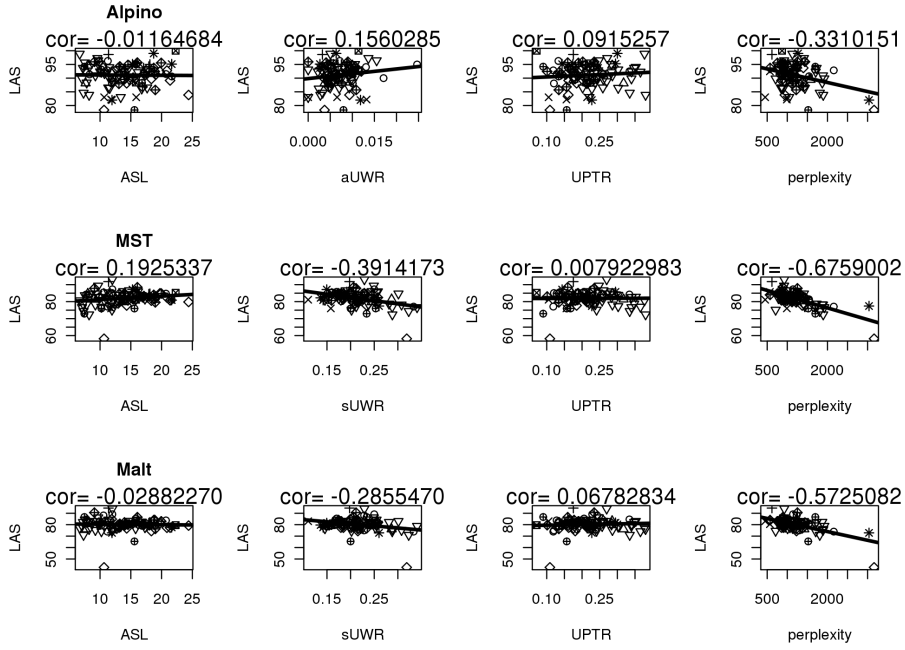


Figure 2: Results: Parser performance on a per-article basis against statistical measure on the text (93 Wikipedia articles - 3 sports articles removed).

Results Figure 2 depicts the correlation between parser performance and the four statistical measures of the text: Average Sentence Length (ASL), simple/Alpino Unknown Word Rate (sUWR/aUWR), Unknown PoS Trigram Rate (UPTR) and perplexity. All parsers are robust to Average Sentence Length (leftmost graphs in Figure 2). They basically do not show any correlation with this measure. This is in line with the results of Zhang and Wang (2009a) for MST and Malt. It is different for the grammar-based parsing system. Their grammar-based parser (ERG/PET) is highly sensitive to Average Sentence Length (correlation coefficient of -0.61 on their datasets), as longer sentences "lead to a sharp drop in parsing coverage of ERG" (Zhang and Wang 2009a). This is not the case for the Alpino parser. The system suffers less from coverage problems and is thus not so sensitive against increasing sentence length.

For Unknown Word Rate (UWR), the data-driven parsers show a high correlation (for this type of task) with this measure (correlation of -0.39 and -0.28),

which is in line with previous findings (Zhang and Wang 2009a). This is not the case for Alpino: again, its very good handling of unknown words makes the system robust to UWR. Note that for Alpino the unknown word rate is measured in a slightly different way (i.e. words not in the lexicon). However, if we would apply the same simple unknown word rate (sUWR) measure to Alpino, it would also result in a weak negative correlation only ($sUWR = -0.07$).

No parser does show any correlation with the third measure, Unknown Part-of-Speech Trigram Rate (UPTR). This is contrary to previous results (Zhang and Wang 2009a), most probably due to the usage of a different tagset and the freer word order language.

Our last measure, perplexity, exhibits the highest correlation to parsing performance: all parsers show the highest sensitivity against this measure, with the data-driven parsers being more sensitive ($cor = -0.67$ and -0.57) than the grammar-driven parser (-0.33). Note that this still holds if we would remove two other possible 'outliers', the diamond and star on the rightmost graphs of Figure 2, resulting in a correlation coefficient of -0.12 (Alpino), -0.57 (MST) and -0.34 (Malt). Moreover, also on DPC (as well as both together; graphs are omitted due to space limits) sentence perplexity gave us the highest correlation to parser performance.[6]

6 Sensitivity of Different Parsing Systems to the Text Domain

We now turn to the second question (Q.2). Clearly, the problem of domain dependence poses a challenge for both kinds of parsing systems, data-driven and grammar-driven. However, to what extent? Which kind of parsing system is more affected by domain shifts? We may rephrase our question as: Which parsing system is more robust to different input texts? To address this issue, we will examine the robustness of the different parsing systems in terms of variation of accuracy on a variety of domains. Note that the goal of this section is not so much to compare individual parser performances, but rather to examine the variability of parser performance across domains.

Towards a measure of domain sensitivity Given a parsing system (p) trained on some source domain and evaluated on a set of N available labeled target domains, the most intuitive measure would be to simply calculate mean (μ) and standard deviation (sd) of the performance on the target domains:

LAS_p^i = accuracy of parser p on target domain i

$$\mu_p^{target} = \frac{\sum_{i=1}^{N} LAS_p^i}{N} \qquad sd_p^{target} = \sqrt{\frac{\sum_{i=1}^{N}(LAS_p^i - \mu_p^{target})^2}{N-1}}$$

However, standard deviation is highly influenced by outliers. Furthermore, this measure does not take the source domain performance (baseline) into consideration nor the size of the target domain itself. We thus propose to measure the do-

[6] One could argue that the cdb corpus might be too small for perplexity scores; however, by using a much larger model estimated by adding the Twente Newspaper Corpus (500 million words) to cdb, the same conclusions are drawn. Perplexity remains the best statistical measure.

main sensitivity of a system, i.e. its *average domain variation* (adv) in accuracy, as weighted average difference from the baseline (source) mean, where the weights represent the size of the various domains:

$$adv = \frac{\sum_{i=1}^{N} w^i * \Delta_p^i}{\sum_{i=1}^{N} w^i} \quad \text{with } \Delta_p^i = LAS_p^i - LAS_p^{baseline},\ w^i = \frac{size(w^i)}{\sum_i^N size(w^i)}$$

In more detail, we propose to measure average domain variation relative to the baseline (source domain) performance by considering non-squared differences from the out-of-domain mean and weigh it by domain size. We thus want the adv measure to take on positive or negative values. Intuitively, to indicate the average weighted gain or loss in performance, relative to the source domain. We will examine this measure in the empirical result section to evaluate the domain sensitivity of the parsers, where $size$ will be measured in terms of number of words (given in Table 8.1). Furthermore, we will measure accuracy per subdomain, not on an article basis, to get more robust statistics.

Baselines To establish our baselines, we perform 5-fold cross validation for each parser on the source domain (cdb corpus, newspaper text). The baselines for each parser are given in Table 8.3.

Model	Alpino	MST	Malt
Baseline (LAS)	90.76	83.63	79.95
Baseline (UAS)	92.47	88.12	83.31

Table 8.3: Baseline (5-fold cross-validation). All differences are significant at $p < 0.001$.

Parser performance across domains As our goal is to assess performance variation across domains, we evaluate each parser on the Wikipedia and DPC corpora that cover a variety of domains (Table 8.1). Figure 3 and Figure 4 summarize the results for each corpus, respectively. In more detail, the figures depict for each parser the baseline performance as given in Table 8.3 (straight lines) and the performance on every domain (bars). Note that domains are ordered by size (number of words), with largest domains on the left. Bars filled with shading lines represent domains in which the parser achieves above-baseline performance, while full-colored bars indicate domains on which the parser's performance falls below source domain baseline, and applying domain adaptation techniques might be fruitful.

Figure 3 depicts parser performance on the Wikipedia domains with respect to the source domain baseline. The figure seems to suggest that the grammar-driven parser Alpino suffers the least from domain shifts. Besides the fact that Alpino scores high overall, its performance is above baseline on several Wikipedia domains. In contrast, the MST parser suffers the most from the domain changes; on most domains a substantial performance drop can be observed. The transition-based parser Malt scores on average lower than the graph-based counterpart, but is less affected by domain shifts than MST and thus lies somewhere in between.

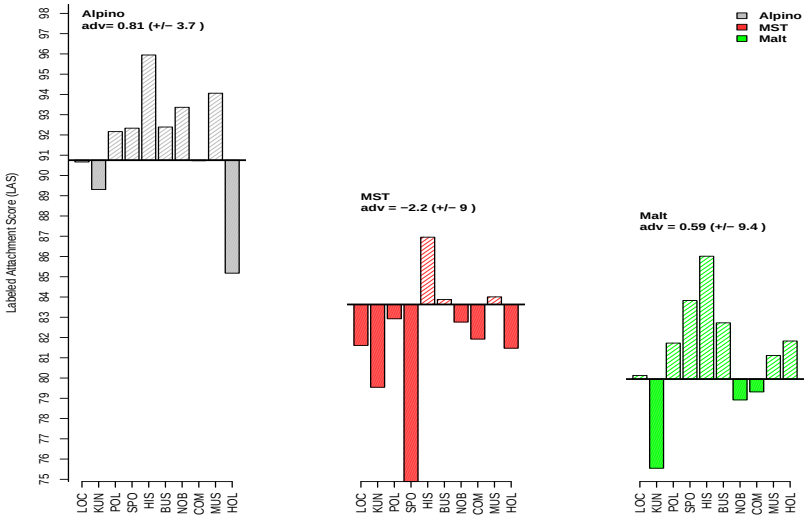


Figure 3: Performance on Wikipedia domains with respect to the source baseline (newspaper text) including average domain variation (adv) score. Domains are ordered by size (largest on left). Full-colored bars indicate domains where performance lies below baseline.

We can summarize these findings by using our proposed average domain variation measure: on average (over all Wikipedia domains), Alpino suffers the least ($adv = +0.81$) - it often scores above baseline, which our measure also suggests (positive score). Alpino is followed by Malt ($adv = +0.59$), also slightly gaining on some domains, and MST ($adv = -2.2$), which on average loses about 2.2 absolute LAS. Thus, MST is clearly the most domain sensitive parser, as also suggested in the graph by the many bars falling below baseline.

The results for the DPC corpus are depicted in Figure 4. It contains a broader set of domains, amongst others science texts (medical texts from the European Medicines Agency as well as texts about oceanography) and articles with more technical vocabulary (Communication, i.e. Internet/ICT texts). Both Malt and Alpino score above baseline on several domains, with this time presumably Malt being slightly less domain affected than Alpino (most probably because Malt scores above baseline on the more influential/larger domains). As with Wikipedia, the figure suggests also here that the MST parser is the most domain-sensitive parser. Our measure supports this finding: MST obtains a negative score ($adv = -0.27$), while Alpino ($adv = 0.22$) and Malt ($adv = 0.4$) achieve on average a gain over the baseline, with Malt being slightly less domain affected

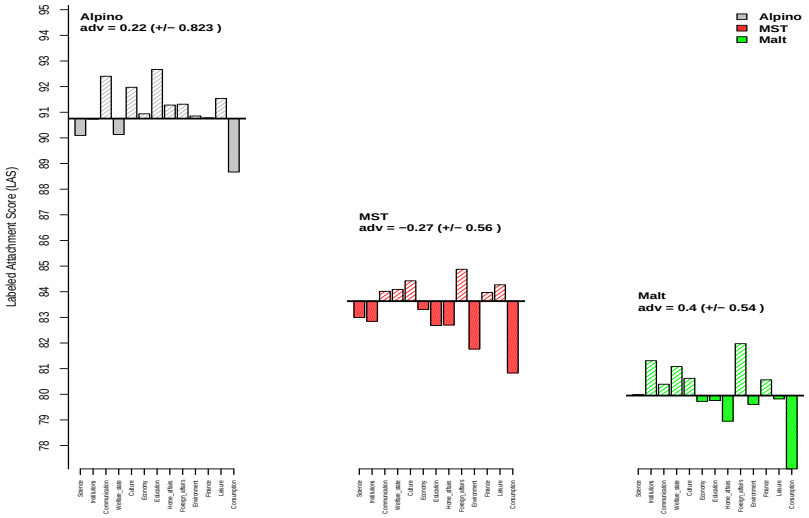


Figure 4: Performance on DPC domains with respect to the source baseline (cdb).

than Alpino. In contrast, if we would take only the deviation on the target domains into consideration (without considering the baseline and the domain size, as discussed in Section 6), we would get a completely opposite ranking on DPC: now the Malt parser would actually be considered the most domain-sensitive (here higher sd means higher sensitivity): Malt ($sd = 1.20$), MST ($sd = 1.14$), Alpino ($sd = 1.05$). However, by looking at Figure 4, intuitively, MST suffers more from the domain shifts than the other parsers, as most bars lie below the baseline. Moreover, the standard deviation measure neither gives a sense of whether the parser on average suffers a loss or gain over the new domains, nor incorporates the information of domain size. We thus believe that our proposed average domain variation is a better suited measure.

To check whether the differences in performance variation are statistically significant, we performed an Approximate Randomization Test over the performance differences (deltas) on the 23 domains (DPC and Wikipedia). The results show that the difference between Alpino and MST is significant. The same goes for the difference between MST and Malt. Thus Alpino is significantly more robust than MST. However, the difference between Alpino and Malt is not significant.

To summarize, our empirical evaluation shows that the grammar-driven system Alpino is rather robust across domains. It is the best performing system and it is significantly more robust than MST. In contrast, the transition-based parser Malt

scores the lowest across all domains, but its variation turned out not to be different from Alpino. Over all domains, MST is the most domain-sensitive parser.

Excursion: Lexical information Both kinds of parsing systems rely on lexical information when learning their parsing (or parse disambiguation) model. However, how much influence does lexical information have? To start examining this issue, we retrain all parsing systems by excluding lexical information. As all systems rely on a feature-based representation, we remove all feature templates that include words or stems and thus train models on a reduced feature space (original versus reduced space: Alpino 24k/7k features; MST 14M/1.9M features; Malt 17/13 templates).

The unlexicalized models are evaluated on the Wikipedia domains. The baseline is again 5-fold cross validation on the source domain (cdb). Obviously, absolute performance drops for all parsers. In more details, lexicalized versus unlexicalized baseline performance in LAS is, for each parser: Alpino $90.75 \rightarrow 89.36$, MST $83.63 \rightarrow 73.14$, Malt $79.95 \rightarrow 73.67$. Thus, as expected, performance drops to a higher degree for the data-driven parsers, but more for MST (-10.49) than for Malt (-6.28). In contrast, the variation in performance across domains (average domain variation) remains similar for most parsers, and is generally slightly smaller (with the exception of Malt): Alpino $adv = 0.81 \rightarrow 0.77$, MST $adv = -2.2 \rightarrow -0.44$, and Malt $adv = 0.59 \rightarrow 1.3$.

From the previous sections we know that the MST parser is the most domain-sensitive parser. The experiment presented in this section seems to suggest that this domain-sensitivity comes indeed from its high reliance on lexical information. When the lexical information is omitted, the MST parser suffers the most: Its absolute performance level drops to 73.14 LAS, even below the unlexicalized baseline of Malt. Moreover, MST scores below Malt on all Wikipedia domains when evaluated in this unlexicalized setting. Thus, even though Malt is the lowest scoring system in the lexicalized case, it seems that Malt is relying less on lexical information, and is thus less affected. In contrast, the grammar-driven parser Alpino suffers less from the missing lexical information.[7]

7 Conclusions and Future Work

We examined a grammar-based system coupled with a statistical disambiguation component (Alpino) and two data-driven statistical parsing systems (MST and Malt) for dependency parsing of Dutch. By looking at the performance variation across a variety of domains, we addressed the question of how sensitive the parsing systems are to the text domain. This, to gauge which kind of system is more affected by domain shifts, and thus more in need for adaptation techniques. We also proposed a new simple measure to quantify domain sensitivity.

The results show that the grammar-based system Alpino is the best performing system, and it is robust across domains. In contrast, MST, the graph-based

[7] Note that Alpino has still access to its lexicon here; for now we removed lexicalized features from the trainable part of Alpino, the statistical disambiguation component.

approach to data-driven parsing is the most domain-sensitive parser. The results for the transition-based parser Malt indicate that its sensitivity is limited, but it is generally (in absolute terms) among the lowest scoring systems. In general, data-driven systems heavily rely on the training data to estimate their models. This becomes apparent when we exclude lexical information from the training process, which results in a substantial performance drop for the data-driven systems, MST and Malt. The grammar-driven model was more robust against the missing lexical information. Grammar-driven systems try to encode domain independent linguistic knowledge, but usually suffer from coverage problems. The Alpino parser successfully implements a set of unknown word heuristics and a partial parsing strategy (in case no full parse can be found) to overcome this problem. This makes the system rather robust across domains, and, as shown in this study, significantly more robust than MST. This is not to say that domain dependence does not consitute a problem for grammar-driven parsers at all. As also noted by Zhang and Wang (2009b), the disambiguation component and lexical coverage of grammar-based systems are still domain-dependent. Thus, domain dependence is a problem for both types of parsing systems, though, as shown in this study, to a lesser extent for the grammar-based system Alpino. Of course, these results are specific for Dutch; however, it's a first step. As the proposed methods are independent of language and parsing system, they can be applied to another system or language.

Another research question examined in this study is how parsing performance correlates to simple statistical measures of the text. By looking at four measures, we could confirm the general result found by Zhang and Wang (2009a): different parsing systems have different sensitivity against statistical measures of the text. While they evaluated parsing systems for English, we here looked at dependency parsing for a freer word order language as Dutch. Both data-driven parsers show a high correlation to unknown word rate, while this is not the case for the grammar-based system. The highest correlation with parsing accuracy was found for the measure we added, sentence perplexity. This is true for both kinds of parsing systems, grammar-based and data-driven, but especially for the statistical parsers MST and Malt. This might first seem counterintuitive, as a grammar-based system usually suffers more from coverage problems. However, as already mentioned, Alpino successfully implements a set of unknown word heuristics to achieve robustness. For instance, on the 'bike winners list' sports domain, which we could identify through these simple statistical measures, Alpino and MST indeed exhibit a very different performance level, showing that the grammar-based system suffered less from the peculiarities of that domain.

In future, we would like to extend this line of work. It might be worth exploring more statistical measures of the text, to build a "domain detection" system for parsing. As for domain sensitivity, we would like to perform an error analysis, to examine why for some domains the parsers outperform their baseline and what are typical in-domain and out-domain errors.

References

Blitzer, John, Mark Dredze, and Fernando Pereira (2007), Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification, *In ACL*, Prague, Czech Republic.

Blitzer, John, Ryan McDonald, and Fernando Pereira (2006), Domain adaptation with structural correspondence learning, *In EMNLP*, Sydney, Australia.

Buchholz, Sabine and Erwin Marsi (2006), Conll-x shared task on multilingual dependency parsing, *In CoNLL*, pp. 149–164.

Daumé III, Hal (2007), Frustratingly easy domain adaptation, *In ACL*, Prague.

Dredze, Mark, John Blitzer, Pratha Pratim Talukdar, Kuzman Ganchev, Joao Graca, and Fernando Pereira (2007), Frustratingly hard domain adaptation for parsing, *Proceedings of the CoNLL Shared Task Session*, Prague, Czech Republic.

Gildea, Daniel (2001), Corpus variation and parser performance, *In EMNLP.*

Hara, Tadayoshi, Miyao Yusuke, and Jun'ichi Tsujii (2005), Adapting a probabilistic disambiguation model of an hpsg parser to a new domain, *Proceedings of the International Joint Conference on Natural Language Processing*.

McClosky, David, Eugene Charniak, and Mark Johnson (2006), Effective self-training for parsing, *In HLT-NAACL*, New York City, pp. 152–159.

McDonald, Ryan, Fernando Pereira, Kiril Ribarov, and Jan Hajič (2005), Non-projective dependency parsing using spanning tree algorithms, *In HLT and EMNLP*, pp. 523–530.

Nivre, Joakim and Ryan McDonald (2008), Integrating graph-based and transition-based dependency parsers, *In HLT-ACL*, Columbus, Ohio, pp. 950–958.

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

Oostdijk, Nelleke (2000), The Spoken Dutch Corpus: Overview and first evaluation, *In LREC*, pp. 887–894.

Plank, Barbara (2010), Improved statistical measures to assess natural language parser performance across domains, *Proceedings of LREC 2010*, Malta.

Plank, Barbara and Gertjan van Noord (2008), Exploring an auxiliary distribution based approach to domain adaptation of a syntactic disambiguation model, *Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation (PE)*, Manchester.

Plank, Barbara and Gertjan van Noord (2010), Grammar-driven versus data-driven: Which parsing system is more affected by domain shifts?, *Proceedings of the 2010 Workshop on NLP and Linguistics: Finding the Common Ground*, Uppsala, Sweden.

Ravi, Sujith, Kevin Knight, and Radu Soricut (2008), Automatic prediction of parser accuracy, *In EMNLP*, Morristown, NJ, USA, pp. 887–896.

Sagae, Kenji, Yusuke Miyao, and Jun'ichi Tsujii (2007), Hpsg parsing with shallow dependency constraints, *In ACL*, Prague, Czech Republic, pp. 624–631.

van Noord, Gertjan (2006), **A**t **L**ast **P**arsing **I**s **N**ow **O**perational, *In TALN 2006 Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues naturelles*, Leuven, pp. 20–42.

Zhang, Yi and Rui Wang (2009a), Correlating natural language parser performance with statistical measures of the text, *In KI 2009*, Germany.

Zhang, Yi and Rui Wang (2009b), Cross-domain dependency parsing using a deep linguistic grammar, *In ACL-IJCNLP*, Suntec, Singapore, pp. 378–386.