Grammar-driven versus Data-driven: Which Parsing System is More Affected by Domain Shifts?

Barbara Plank
University of Groningen
The Netherlands
b.plank@rug.nl

Gertjan van Noord
University of Groningen
The Netherlands
G.J.M.van.Noord@rug.nl

Abstract

In the past decade several parsing systems for natural language have emerged, which use different methods and formalisms. For instance, systems that employ a hand-crafted grammar and 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? Intuitively, grammar-driven systems should be less affected by domain changes. To investigate this hypothesis, an empirical investigation on Dutch is carried out. The performance variation of a grammar-driven versus two data-driven systems across domains is evaluated, and a simple measure to quantify domain sensitivity proposed. This will give an estimate of which parsing system is more affected by domain shifts, and thus more in need for adaptation techniques.

1 Introduction

Most modern Natural Language Processing (NLP) systems are subject to the wellknown problem of 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 the 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 clearly 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 Analysis (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 grammar. Yet, domain adaptation poses a challenge for both kinds of parsing systems. But to what extent do these different kinds of systems suffer from the problem? We test the hypothesis that grammar-driven systems are less affected by domain changes. We empirically investigate this in a case-study on Dutch.

Proceedings of the 2010 Workshop on NLP and Linguistics: Finding the Common Ground, ACL 2010, pages 25–33,
Uppsala, Sweden, 16 July 2010. ©2010 Association for Computational Linguistics

2 Related work

Most previous work has focused on a single parsing system in isolation (Gildea, 2001; Hara et al., 2005; McClosky et al., 2006). However, 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 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 (2009) 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. This is the work most closest to ours.

However, which kind of system (hand-crafted versus purely statistical) is more affected by the domain, and thus more sensitive to domain shifts? To the best of our knowledge, no study has yet addressed this issue. We thus assess the performance variation of three dependency parsing systems for Dutch across domains, and propose a simple measure to quantify domain sensitivity.

3 Parsing Systems

The parsing systems used in this study are: a grammar-based system for Dutch (Alpino) and two data-driven systems (MST and Malt), all described next.

(1) Alpino is a parser for Dutch which has been developed over the last ten years, on the basis of a domain-specific HPSG-grammar that was used in the OVIS spoken dialogue system. The OVIS parser was shown to out-perform a statistical (DOP) parser, in a contrastive formal evaluation (van Zanten et al., 1999). In the ten years after this evaluation, the system has developed into a generic parser for Dutch. Alpino consists of more than 800 grammar rules in the tradition of HPSG, and a large hand-crafted lexicon. It produces dependency structures as ouput, where more than a single head per token is allowed. For words that are not in the lexicon, the system applies a large variety of unknown word heuristics (van Noord, 2006), which deal with number-like expressions, compounds, proper names, etc. Coverage of the grammar and lexicon has been extended over the years by paying careful attention to the results of parsing large corpora, by means of error mining techniques (van Noord, 2004; de Kok et al., 2009).

Lexical ambiguity is reduced by means of a POS-tagger, described in (Prins and van Noord, 2003). This POS-tagger is trained on large amounts of parser output, and removes unlikely lexical categories. Some amount of lexical ambiguity remains. A left-corner parser constructs a parse-forest for an input sentence. Based on large amounts of parsed data, the parser considers only promising parse step sequences, by filtering out sequences of parse steps which were not previously used to construct a best parse for a given sentence. The parse step filter improves efficiency considerably (van Noord, 2009).

A best-first beam-search algorithm retrieves the best parse(s) from that forest by consulting a Maximum Entropy disambiguation component. Features for the disambiguation component include non-local features. For instance, there are features that can be used to learn a preference for local extraction over long-distance extraction, and a preference for subject fronting rather than direct object fronting, and a preference for certain types of orderings in the "mittelfeld" of a Dutch sentence. The various features that we use for disambiguation, as well as the best-first algorithm is described in (van Noord, 2006). The model now also contains features which implement selection restrictions, trained on the basis of large parsed corpora (van Noord, 2007). The maximum entropy disambiguation component is trained on the Alpino treebank, described below.

To illustrate the role of the disambiguation component, we provide some results for the first 536 sentences of one of the folds of the training data (of course, the model used in this experiment is trained on the remaining folds of training data). In this setup, the POS-tagger and parse step filter already filter out many, presumably bad, parses. This table indicates that a very large amount of parses can be constructed for some sentences. Furthermore, the maximum entropy disambiguation component does a good job in selecting good parses from those. Accuracy is given here in terms of f-score of named dependencies.

sents	parses	oracle	arbitrary	model
536	45011	95.74	76.56	89.39

(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. 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 Alpino Treebank (van Noord, 2006). For our cross-domain evaluation, we consider Wikipedia and DPC (Dutch Parallel Corpus) as target data. All datasets are described next.

Source: Cdb The cdb (Alpino Treebank) consists of 140,000 words (7,136 sentences) from the Eindhoven corpus (newspaper text). It is a collection of text fragments from 6 Dutch newspapers. The collection has been annotated according to the guidelines of CGN (Oostdijk, 2000) and stored in XML format. 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 training data additionally contained a mix of non-newspaper text,[1] which we exclude here on purpose to keep a clean baseline.

Target: Wikipedia and DPC We use the Wikipedia and DPC subpart of the LASSY corpus[2] as target domains. These corpora contain several domains, e.g. sports, locations, science. On overview of the corpora is given in Table 1. Note that both consist of hand-corrected data labeled by Alpino, thus all domains employ the same annotation scheme. This might introduce a slight bias towards Alpino, however it has the advantage that all domains employ the same annotation scheme – which was the major source of error in the CoNLL task on domain adaptation (Dredze et al., 2007).

Wikipedia	**Example articles**	**#a**	**#w**	**ASL**
LOC (location)	Belgium, Antwerp (city)	31	25259	11.5
KUN (arts)	Tervuren school	11	17073	17.1
POL (politics)	Belgium elections 2003	16	15107	15.4
SPO (sports)	Kim Clijsters	9	9713	11.1
HIS (history)	History of Belgium	3	8396	17.9
BUS (business)	Belgium Labour Federation	9	4440	11.0
NOB (nobility)	Albert II	6	4179	15.1
COM (comics)	Suske and Wiske	3	4000	10.5
MUS (music)	Sandra Kim, Urbanus	3	1296	14.6
HOL (holidays)	Flemish Community Day	4	524	12.2
Total		**95**	**89987**	**13.4**

DPC	**Description/Example**	**#a**	**#words**	**ASL**
Science	medicine, oeanography	69	60787	19.2
Institutions	political speeches	21	28646	16.1
Communication	ICT/Internet	29	26640	17.5
Welfare state	pensions	22	20198	17.9
Culture	darwinism	11	16237	20.5
Economy	inflation	9	14722	18.5
Education	education in Flancers	2	11980	16.3
Home affairs	presentation (Brussel)	1	9340	17.3
Foreign affairs	European Union	7	9007	24.2
Environment	threats/nature	6	8534	20.4
Finance	banks (education banker)	6	6127	22.3
Leisure	various (drugscandal)	2	2843	20.3
Consumption	toys from China	1	1310	22.6
Total		**186**	**216371**	**18.5**

Table 1: Overview Wikipedia and DPC corpus (#a articles, #w words, ASL average sentence length)

CoNLL2006 This is the testfile for Dutch that was used in the CoNLL 2006 shared task on multilingual dependency parsing. The file consists of 386 sentences from an institutional brochure (about youth healthcare). We use this file to check our data-driven models against state-of-the-art.

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 multilingual dependency parsing. Instead of using the PoS tagger and tagset used in the shared task (to which we did not have access to), we replaced the PoS tags with more fine-grained tags obtained by

[1] Namely, a large amount of questions (from CLEF, roughly 4k sentences) and hand-crafted sentences used during the development of the grammar (1.5k).

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

parsing the data with the Alpino parser.[3] At testing time, the data-driven parsers are given PoS tagged input, while Alpino gets 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 evaluation 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 would be 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 ('secondary edges') (van Noord, 2006). However, such edges are ignored in the CoNLL format; just a single head per token is allowed. 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. To determine whether results are significant, we us the *Approximate Randomization Test* (see Yeh (2000)) with 1000 random shuffles.

5 Domain sensitivity

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 answer this question, we will examine the robustness of the different parsing systems in terms of variation of accuracy on a variety of domains.

A measure of domain sensitivity Given a parsing system (p) trained on some source domain and evaluated on a set of N 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

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

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

$$adv = \frac{\sum_{i=1}^{N} w^i * \Delta_p^i}{\sum_{i=1}^{N} w^i}, \text{ with}$$

$$\Delta_p^i = LAS_p^i - LAS_p^{baseline} \text{ and } w^i = \frac{size(w^i)}{\sum_{i=1}^{N} size(w^i)}$$

In more detail, we measure *average domain variation* (adv) relative to the baseline (source domain) performance by considering non-squared differences from the out-of-domain mean and weigh it by domain size. The adv measure can thus take on positive or negative values. Intuitively, it will indicate the average weighted gain or loss in performance, relative to the source domain. As alternative, we may want to just calculate a straight, unweighted average: $uadv = \sum_{i=1}^{N} \Delta_p^i / N$. However, this assumes that domains have a representative size, and a threshold might be needed to disregard domains that are presumably too small.

We will use adv in the empirical result section to evaluate the domain sensitivity of the parsers, where $size$ will be measured in terms of number of words. We additionally provide values for the unweighted version using domains with at least 4000 words (cf. Table 1).

6 Empirical results

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 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.

[3] As discussed later (Section 6, cf. Table 2), using Alpino tags actually improves the performance of the data-driven parsers. 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).

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

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
Malt (cdb retagged with Mbt)	75.34	78.29

Table 2: Performance of data-driven parsers versus state-of-the-art on the CoNLL 2006 testset (in Labeled/Unlabeled Attachment Score).

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 2, by using Alpino tags the performance level significantly improves (with $p < 0.002$ using 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 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). Thus, 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.

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 3. The grammar-driven parser Alpino achieves a baseline that is significantly higher (90.75% LAS) compared to the baselines of the data-driven systems (around 80-83% LAS).

Cross-domain results 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 (described in Table 1). Figure 1 and Figure 2 summarizes the results for each corpus, respectively. In more detail, the figures depict for each parser the baseline performance as given in Table 3 (straight lines) and the performance on every domain (bars). Note that domains are ordered by size (number of words), so that the largest domains appear as bars on the left. Similar graphs come up if we replace labeled attachment score with its unlabeled variant.

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

Table 3: Baseline (5-fold cross-validation). All differences are significant at $p < 0.001$.

Figure 1 depicts parser performance on the Wikipedia domains with respect to the source domain baseline. The figure indicates that the grammar-driven parser does not suffer much from domain shifts. Its performance falls even above baseline for 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 scores on average significantly lower than the graph-based counterpart and Alpino, but seems to be less affected by the domain shifts.

We can summarize this findings by our proposed average domain variation measure (unweighted scores are given in the Figure): On average (over all Wikipedia domains), Alpino suffers the least ($adv = +0.81$), followed by Malt ($+0.59$) and MST (-2.2), which on average loses 2.2 absolute LAS. Thus, the graph-based data-driven dependency parser MST suffers the most.

We evaluate the parsers also on the more varied DPC corpus. 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). The results are depicted in Figure 2. Both Malt ($adv = 0.4$) and Alpino ($adv = 0.22$) achieve on average a gain over the baseline, with this time Malt being slightly less domain affected than Alpino (most probably because Malt scores above average on the more influential/larger domains). Nevertheless, Alpino's performance level is significantly higher compared to both data-driven counterparts. The graph-based data-driven

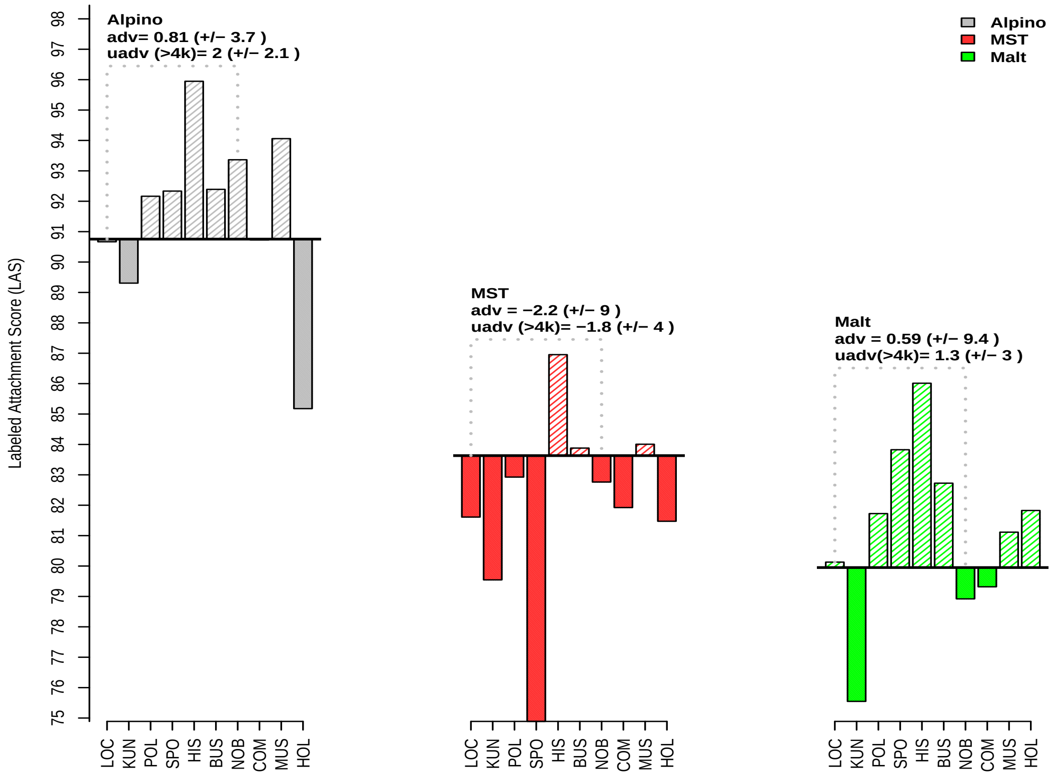

Figure 1: Performance on Wikipedia domains with respect to the source baseline (newspaper text) including average domain variation (adv) score and its unweighted alternative (uadv). Domains are ordered by size (largest on left). Full-colored bars indicate domains where performance lies below the baseline.

parser MST is the most domain-sensitive parser also on DPC ($adv = -0.27$).

In contrast, if we would take only the deviation on the target domains into consideration (without considering the baseline, cf. Section 5), we would get a completely opposite ranking on DPC, where 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 2, intuitively, MST suffers more from the domain shifts than Malt, 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 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. These findings hold for differences measured in both labeled and unlabeled attachments scores. Furthermore, all differences in absolute performance across domains are 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 constrast, 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.

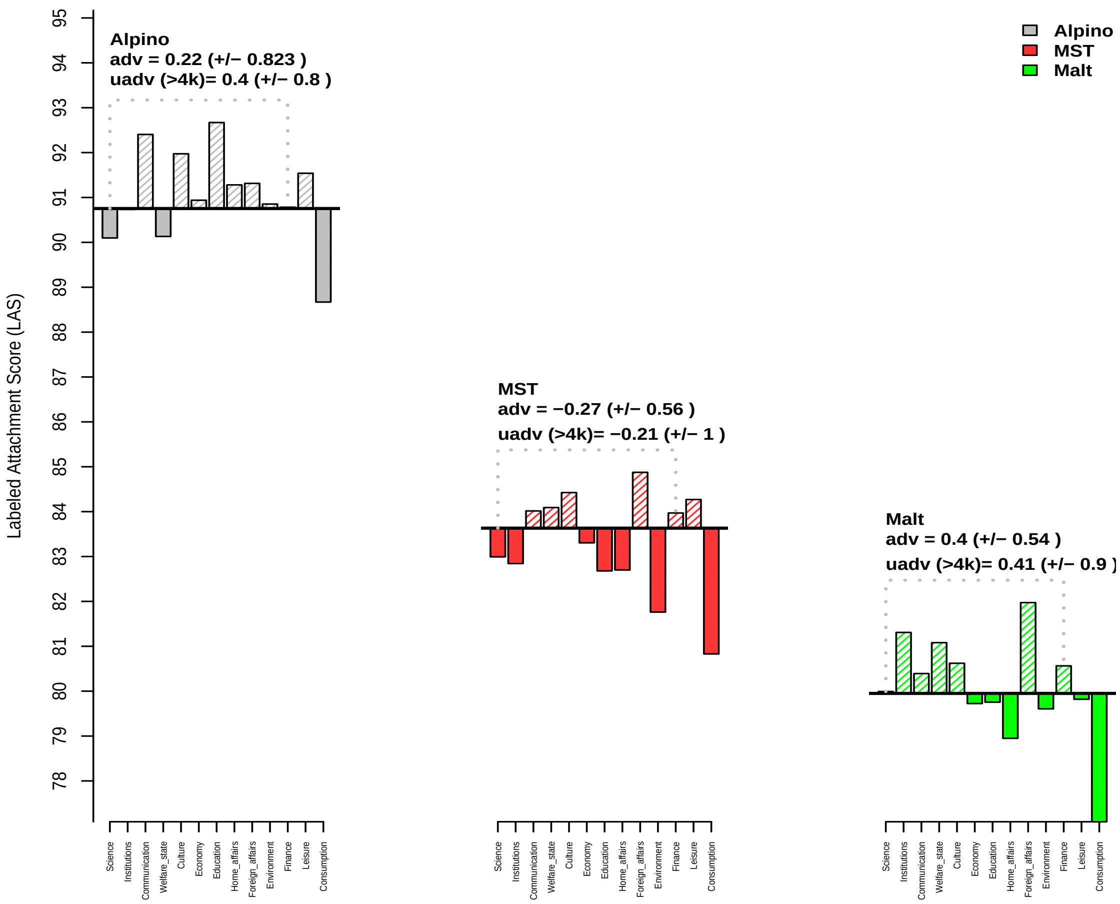

Figure 2: Performance on DPC domains with respect to the source baseline (newspaper text).

Excursion: Lexical information Both kinds of parsing systems rely on lexical information (words/stems) when learning their parsing (or parse disambiguation) model. However, how much influence does lexical information have?

To examine this issue, we retrain all parsing systems by excluding lexical information. As all parsing systems rely on a feature-based representation, we remove all feature templates that include words 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 result of evaluating the unlexicaled models on Wikipedia are shown in Figure 3. Clearly, performance drops for for all parsers in all domains. However, for the data-driven parsers to a much higher degree. For instance, MST loses on average 11 absolute points in performance ($adv = -11$) and scores below baseline on all Wikipedia domains. In contrast, the grammar-driven parser Alpino suffers far less, still scores above baseline on some domains.[5] The Malt parser lies somewhere in between, also suffers from the missing lexical information, but to a lesser degree than the graph-based parser MST.

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 large variety of domains, we addressed the question of how sensitive the parsing systems are to the text domain. This, to gauge which kind

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

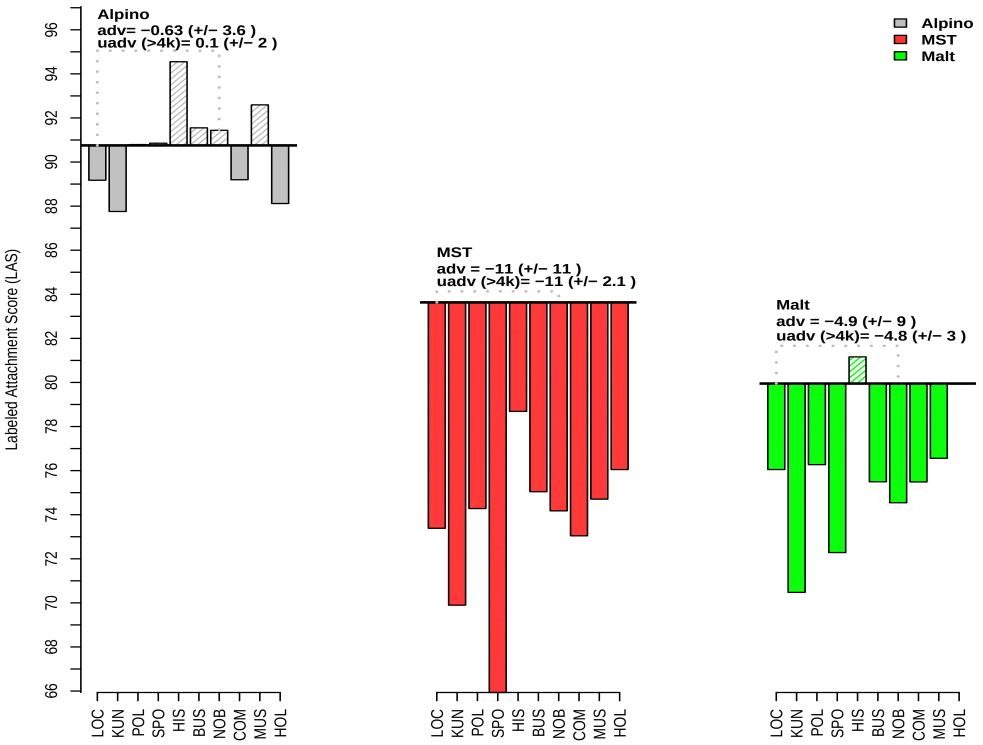

Figure 3: Performance of unlexical parsers on Wikipedia domains with respect to the source baseline.

of system (data-driven versus grammar-driven) is more affected by domain shifts, and thus more in need for adaptation techniques. We also proposed a 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 approach to data-driven parsing is the most domain-sensitive parser. The results for Malt indicate that its variation across domains is limited, but this parser is outperformed by both other systems on all domains. 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 (2009), 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 indepedent of language and parsing system, they can be applied to another system or language.

In future, we would like to (a) perform an error analysis (e.g. why for some domains the parsers outperform their baseline; what are typical in-domain and out-domain errors), (a) examine why there is such a difference in performance variation between Malt and MST, and (c) investigate what part(s) of the Alpino parser are responsible for the differences with the data-driven parsers.

References

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Conference on Empirical Methods in Natural Language Processing*, Sydney.

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

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *In Proc. of CoNLL*, pages 149–164.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *ACL*, Prague, Czech Republic.

Daniël de Kok, Jianqiang Ma, and Gertjan van Noord. 2009. A generalized method for iterative error mining in parsing results. In *Proceedings of the 2009 Workshop on Grammar Engineering Across Frameworks (GEAF 2009)*, pages 71–79, Suntec, Singapore, August.

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

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP).*

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

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-projective dependency parsing using spanning tree algorithms. In *In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 523–530.

Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of ACL-08: HLT*, pages 950–958, Columbus, Ohio, June.

Joakim Nivre, 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*, 13:95–135.

Nelleke Oostdijk. 2000. The Spoken Dutch Corpus: Overview and first evaluation. In *Proceedings of LREC*, pages 887–894.

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

Robbert Prins and Gertjan van Noord. 2003. Reinforcing parser preferences through tagging. *Traitement Automatique des Langues*, 44(3):121–139.

Kenji Sagae, Yusuke Miyao, and Jun'ichi Tsujii. 2007. Hpsg parsing with shallow dependency constraints. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 624–631, Prague, Czech Republic, June.

Gertjan van Noord. 2004. Error mining for wide-coverage grammar engineering. In *ACL2004*, Barcelona. ACL.

Gertjan van Noord. 2006. **At** **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*, pages 20–42, Leuven.

Gertjan van Noord. 2007. Using self-trained bilexical preferences to improve disambiguation accuracy. In *Proceedings of the International Workshop on Parsing Technology (IWPT)*, ACL 2007 Workshop, pages 1–10, Prague. ACL.

Gertjan van Noord. 2009. Learning efficient parsing. In *EACL 2009, The 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 817–825, Athens, Greece.

Gert Veldhuijzen van Zanten, Gosse Bouma, Khalil Sima'an, Gertjan van Noord, and Remko Bonnema. 1999. Evaluation of the NLP components of the OVIS2 spoken dialogue system. In Frank van Eynde, Ineke Schuurman, and Ness Schelkens, editors, *Computational Linguistics in the Netherlands 1998*, pages 213–229. Rodopi Amsterdam.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *ACL*, pages 947–953, Morristown, NJ, USA.

Yi Zhang and Rui Wang. 2009. Cross-domain dependency parsing using a deep linguistic grammar. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 378–386, Suntec, Singapore, August.