Exploring an Auxiliary Distribution based approach to Domain Adaptation of a Syntactic Disambiguation Model

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

We investigate auxiliary distributions (Johnson and Riezler, 2000) for domain adaptation of a supervised parsing system of Dutch. To overcome the limited target domain training data, we exploit an original and larger out-of-domain model as auxiliary distribution. However, our empirical results exhibit that the auxiliary distribution does not help: even when very little target training data is available the incorporation of the out-of-domain model does not contribute to parsing accuracy on the target domain; instead, better results are achieved either without adaptation or by simple model combination.

1 Introduction

Modern statistical parsers are trained on large annotated corpora (treebanks) and their parameters are estimated to reflect properties of the training data. Therefore, a disambiguation component will be successful as long as the treebank it was trained on is representative for the input the model gets. However, as soon as the model is applied to another *domain*, or *text genre* (Lease et al., 2006), accuracy degrades considerably. For example, the performance of a parser trained on the Wall Street Journal (newspaper text) significantly drops when evaluated on the more varied Brown (fiction/non-fiction) corpus (Gildea, 2001).

A simple solution to improve performance on a new domain is to construct a parser specifically for that domain. However, this amounts to hand-labeling a considerable amount of training data which is clearly very expensive and leads to an unsatisfactory solution. In alternative, techniques for *domain adaptation*, also known as *parser adaptation* (McClosky et al., 2006) or *genre portability* (Lease et al., 2006), try to leverage either a small amount of already existing annotated data (Hara et al., 2005) or unlabeled data (McClosky et al., 2006) of one domain to parse data from a different domain. In this study we examine an approach that assumes a limited amount of already annotated in-domain data.

We explore auxiliary distributions (Johnson and Riezler, 2000) for domain adaptation, originally suggested for the incorporation of lexical selectional preferences into a parsing system. We gauge the effect of exploiting a more general, out-of-domain model for parser adaptation to overcome the limited amount of in-domain training data. The approach is examined on two application domains, question answering and spoken data.

For the empirical trials, we use Alpino (van Noord and Malouf, 2005; van Noord, 2006), a robust computational analyzer for Dutch. Alpino employs a discriminative approach to parse selection that bases its decision on a Maximum Entropy (MaxEnt) model. Section 2 introduces the MaxEnt framework. Section 3 describes our approach of exploring auxiliary distributions for domain adaptation. In section 4 the experimental design and empirical results are presented and discussed.

2 Background: MaxEnt Models

Maximum Entropy (MaxEnt) models are widely used in Natural Language Processing (Berger et al., 1996; Ratnaparkhi, 1997; Abney, 1997). In this framework, a disambiguation model is speci-

Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation, pages 9–16
Manchester, August 2008

fied by a set of feature functions describing properties of the data, together with their associated weights. The weights are learned during the training procedure so that their estimated value determines the contribution of each feature. In the task of parsing, features appearing in correct parses are given increasing weight, while features in incorrect parses are given decreasing weight. Once a model is trained, it can be applied to parse selection that chooses the parse with the highest sum of feature weights.

During the training procedure, the weights vector is estimated to best fit the training data. In more detail, given m features with their corresponding empirical expectation $E_{\tilde{p}}[f_j]$ and a default model q_0, we seek a model p that has minimum Kullback-Leibler (KL) divergence from the default model q_0, subject to the expected-value constraints: $E_p[f_j] = E_{\tilde{p}}[f_j]$, where $j \in 1, ..., m$.

In MaxEnt estimation, the default model q_0 is often only implicit (Velldal and Oepen, 2005) and not stated in the model equation, since the model is assumed to be uniform (e.g. the constant function $\frac{1}{\Omega(s)}$ for sentence s, where $\Omega(s)$ is the set of parse trees associated with s). Thus, we seek the model with minimum KL divergence from the uniform distribution, which means we search model p with maximum entropy (uncertainty) subject to given constraints (Abney, 1997).

In alternative, if q_0 is not uniform then p is called a *minimum divergence model* (according to (Berger and Printz, 1998)). In the statistical parsing literature, the default model q_0 that can be used to incorporate prior knowledge is also referred to as base model (Berger and Printz, 1998), default or reference distribution (Hara et al., 2005; Johnson et al., 1999; Velldal and Oepen, 2005).

The solution to the estimation problem of finding distribution p, that satisfies the expected-value constraints and minimally diverges from q_0, has been shown to take a specific parametric form (Berger and Printz, 1998):

$$p_\theta(\omega, s) = \frac{1}{Z_\theta} q_0 exp^{\sum_{j=1}^{m} \theta_j f_j(\omega)} \quad (1)$$

with m feature functions, s being the input sentence, ω a corresponding parse tree, and Z_θ the normalization equation:

$$Z_\theta = \sum_{\omega' \in \Omega} q_0 exp^{\sum_{j=1}^{m} \theta_j f_j(\omega')} \quad (2)$$

Since the sum in equation 2 ranges over all possible parse trees $\omega' \in \Omega$ admitted by the grammar, calculating the normalization constant renders the estimation process expensive or even intractable (Johnson et al., 1999). To tackle this problem, Johnson et al. (1999) redefine the estimation procedure by considering the conditional rather than the joint probability.

$$P_\theta(\omega|s) = \frac{1}{Z_\theta} q_0 exp^{\sum_{j=1}^{m} \theta_j f_j(\omega)} \quad (3)$$

with Z_θ as in equation 2, but instead, summing over $\omega' \in \Omega(s)$, where $\Omega(s)$ is the set of parse trees associated with sentence s. Thus, the probability of a parse tree is estimated by summing only over the possible parses of a specific sentence.

Still, calculating $\Omega(s)$ is computationally very expensive (Osborne, 2000), because the number of parses is in the worst case exponential with respect to sentence length. Therefore, Osborne (2000) proposes a solution based on *informative samples*. He shows that is suffices to train on an informative subset of available training data to accurately estimate the model parameters. Alpino implements the Osborne-style approach to Maximum Entropy parsing. The standard version of the Alpino parser is trained on the Alpino newspaper Treebank (van Noord, 2006).

3 Exploring auxiliary distributions for domain adaptation

3.1 Auxiliary distributions

Auxiliary distributions (Johnson and Riezler, 2000) offer the possibility to incorporate information from additional sources into a MaxEnt Model. In more detail, auxiliary distributions are integrated by considering the logarithm of the probability given by an auxiliary distribution as an additional, real-valued feature. More formally, given k auxiliary distributions $Q_i(\omega)$, then k new *auxiliary features* $f_{m+1}, ..., f_{m+k}$ are added such that

$$f_{m+i}(\omega) = \log Q_i(\omega) \quad (4)$$

where $Q_i(\omega)$ do not need to be proper probability distributions, however they must strictly be positive $\forall \omega \in \Omega$ (Johnson and Riezler, 2000).
The auxiliary distributions resemble a reference distribution, but instead of considering a single reference distribution they have the advantage that several auxiliary distributions can be integrated and weighted against each other. John-

son establishes the following equivalence between the two (Johnson and Riezler, 2000; Velldal and Oepen, 2005):

$$Q(\omega) = \prod_{i=1}^{k} Q_i(\omega)^{\theta_{m+i}} \quad (5)$$

where $Q(\omega)$ is the reference distribution and $Q_i(\omega)$ is an auxiliary distribution. Hence, the contribution of each auxiliary distribution is regulated through the estimated feature weight. In general, a model that includes k auxiliary features as given in equation (4) takes the following form (Johnson and Riezler, 2000):

$$P_\theta(\omega|s) = \frac{\prod_{i=1}^{k} Q_i(\omega)^{\theta_{m+i}}}{Z_\theta} exp^{\sum_{j=1}^{m} \theta_j f_j(\omega)} \quad (6)$$

Due to the equivalence relation in equation (5) we can restate the equation to explicitly show that auxiliary distributions are additional features[1].

$$P_\theta(\omega|s)$$

$$= \frac{\prod_{i=1}^{k} [exp^{f_{m+i(\omega)}}]^{\theta_{m+i}}}{Z_\theta} exp^{\sum_{j=1}^{m} \theta_j f_j(\omega)} \quad (7)$$

$$= \frac{1}{Z_\theta} \prod_{i=1}^{k} exp^{f_{m+i(\omega)} * \theta_{m+i}} exp^{\sum_{j=1}^{m} \theta_j f_j(\omega)} \quad (8)$$

$$= \frac{1}{Z_\theta} exp^{\sum_{i=1}^{k} f_{m+i(\omega)} * \theta_{m+i}} exp^{\sum_{j=1}^{m} \theta_j f_j(\omega)} \quad (9)$$

$$= \frac{1}{Z_\theta} exp^{\sum_{j=1}^{m+k} \theta_j f_j(\omega)}$$
$$\text{with } f_j(\omega) = logQ(\omega) \text{ for } m < j \leq (m+k) \quad (10)$$

3.2 Auxiliary distributions for adaptation

While (Johnson and Riezler, 2000; van Noord, 2007) focus on incorporating several auxiliary distributions for lexical selectional preferences, in this study we explore auxiliary distributions for domain adaptation.

We exploit the information of the more general model, estimated from a larger, out-of-domain treebank, for parsing data from a particular target domain, where only a small amount of training data is available. A related study is Hara et al. (2005). While they also assume a limited amount of in-domain training data, their approach differs from ours in that they incorporate an original model as a reference distribution, and their estimation procedure is based on parse forests (Hara et al., 2005; van Noord, 2006), rather than informative samples. In this study, we want to gauge the effect of auxiliary distributions, which have the advantage that the contribution of the additional source is regulated.

More specifically, we extend the target model to include (besides the original integer-valued features) one additional real-valued feature (k=1)[2]. Its value is defined to be the negative logarithm of the conditional probability given by OUT, the original, out-of-domain, Alpino model. Hence, the general model is 'merged' into a single auxiliary feature:

$$f_{m+1} = -logP_{OUT}(\omega|s) \quad (11)$$

The parameter of the new feature is estimated using the same estimation procedure as for the remaining model parameters. Intuitively, our auxiliary feature models dispreferences of the general model for certain parse trees. When the Alpino model assigns a high probability to a parse candidate, the auxiliary feature value will be small, close to zero. In contrast, a low probability parse tree in the general model gets a higher feature value. Together with the estimated feature weight expected to be negative, this has the effect that a low probability parse in the Alpino model will reduce the probability of a parse in the target domain.

3.3 Model combination

In this section we sketch an alternative approach where we keep only two features under the MaxEnt framework: one is the log probability assigned by the out-domain model, the other the log probability assigned by the in-domain model:

$$f_1 = -logP_{OUT}(\omega|s), f_2 = -logP_{IN}(\omega|s)$$

The contribution of each feature is again scaled through the estimated feature weights θ_1, θ_2.
We can see this as a simple instantiation of *model combination*. In alternative, *data combination* is a domain adaptation method where IN and OUT-domain data is simply concatenated and a new model trained on the union of data. A potential and well known disadvantage of data combination is that the usually larger amount of out-domain data

[1] Note that the step from equation (6) to (7) holds by restating equation (4) as $Q_i(\omega) = exp^{f_{m+i}(\omega)}$

[2] Or alternatively, $k \geq 1$ (see section 4.3.1).

'overwhelms' the small amount of in-domain data. Instead, Model combination interpolates the two *models* in a linear fashion by scaling their contribution. Note that if we skip the parameter estimation step and simply assign the two parameters equal values (equal weights), the method reduces to $P_{OUT}(\omega|s) \times P_{IN}(\omega|s)$, i.e. just multiplying the respective model probabilities.

4 Experiments and Results

4.1 Experimental design

The general model is trained on the Alpino Treebank (van Noord, 2006) (newspaper text; approximately 7,000 sentences). For the domain-specific corpora, in the first set of experiments (section 4.3) we consider the Alpino CLEF Treebank (questions; approximately 1,800 sentences). In the second part (section 4.4) we evaluate the approach on the Spoken Dutch corpus (Oostdijk, 2000) (CGN, 'Corpus Gesproken Nederlands'; spoken data; size varies, ranging from 17 to 1,193 sentences). The CGN corpus contains a variety of components/subdomains to account for the various dimensions of language use (Oostdijk, 2000).

4.2 Evaluation metric

The output of the parser is evaluated by comparing the generated dependency structure for a corpus sentence to the gold standard dependency structure in a treebank. For this comparison, we represent the dependency structure (a directed acyclic graph) as a set of named dependency relations. To compare such sets of dependency relations, we count the number of dependencies that are identical in the generated parse and the stored structure, which is expressed traditionally using precision, recall and f-score (Briscoe et al., 2002).

Let D_p^i be the number of dependencies produced by the parser for sentence i, D_g^i is the number of dependencies in the treebank parse, and D_o^i is the number of correct dependencies produced by the parser. If no superscript is used, we aggregate over all sentences of the test set, i.e.,:

$$D_p = \sum_i D_p^i \qquad D_o = \sum_i D_o^i \qquad D_g = \sum_i D_g^i$$

Precision is the total number of correct dependencies returned by the parser, divided by the overall number of dependencies returned by the parser (precision $= D_o/D_p$); recall is the number of correct system dependencies divided by the total number of dependencies in the treebank (recall $= D_o/D_g$). As usual, precision and recall can be combined in a single f-score metric.

An alternative similarity score for dependency structures is based on the observation that for a given sentence of n words, a parser would be expected to return n dependencies. In such cases, we can simply use the percentage of correct dependencies as a measure of accuracy. Such a labeled dependency accuracy is used, for instance, in the CoNLL shared task on dependency parsing ("labeled attachment score").

Our evaluation metric is a variant of labeled dependency accuracy, in which we do allow for some discrepancy between the number of returned dependencies. Such a discrepancy can occur, for instance, because in the syntactic annotations of Alpino (inherited from the CGN) words can sometimes be dependent on more than a single head (called 'secondary edges' in CGN). A further cause is parsing failure, in which case a parser might not produce any dependencies. We argue elsewhere (van Noord, In preparation) that a metric based on f-score can be misleading in such cases. The resulting metric is called *concept accuracy*, in, for instance, Boros et al. (1996).[3]

$$\text{CA} = \frac{D_o}{\sum_i \max(D_g^i, D_p^i)}$$

The concept accuracy metric can be characterized as the mean of a per-sentence minimum of recall and precision. The resulting CA score therefore is typically slightly lower than the corresponding f-score, and, for the purposes of this paper, equivalent to labeled dependency accuracy.

4.3 Experiments with the QA data

In the first set of experiments we focus on the Question Answering (QA) domain (CLEF corpus). Besides evaluating our auxiliary based approach (section 3), we conduct separate baseline experiments:

- **In-domain (CLEF):** train on CLEF (baseline)
- **Out-domain (Alpino):** train on Alpino
- **Data Combination (CLEF+Alpino):** train a model on the combination of data, CLEF ∪ Alpino

[3] In previous publications and implementations definitions were sometimes used that are equivalent to: CA $= \frac{D_o}{\max(D_g, D_p)}$ which is slightly different; in practice the differences can be ignored.

Dataset size (#sents)	In-dom. CLEF	Out-dom. Alpino	Data Combination CLEF+Alpino	Aux.distribution CLEF+Alpino_aux	Model Combination CLEF_aux+Alpino_aux	Model Combination equal weights
CLEF 2003 (446)	97.01	94.02	97.21	97.01	97.14	97.46
CLEF 2004 (700)	96.60	89.88	95.14	96.60	97.12	97.23
CLEF 2005 (200)	97.65	87.98	93.62	97.72	97.99	98.19
CLEF 2006 (200)	97.06	88.92	95.16	97.06	97.00	96.45
CLEF 2007 (200)	96.20	92.48	97.30	96.33	96.33	96.46

Table 1: Results on the CLEF test data; underlined scores indicate results $>$ in-domain baseline (CLEF)

- **Auxiliary distribution (CLEF+Alpino_aux):** adding the original Alpino model as auxiliary feature to CLEF
- **Model Combination:** keep only two features $P_{OUT}(\omega|s)$ and $P_{IN}(\omega|s)$. Two variants: i) estimate the parameters θ_1, θ_2 (**CLEF_aux+Alpino_aux**); ii) give them equal values, i.e. θ_1=θ_2=-1 (**equal weights**)

We assess the performance of all of these models on the CLEF data by using 5-fold cross-validation. The results are given in table 1.

The CLEF model performs significantly better than the out-of-domain (Alpino) model, despite of the smaller size of the in-domain training data. In contrast, the simple data combination results in a model (CLEF+Alpino) whose performance is somewhere in between. It is able to contribute in some cases to disambiguate questions, while leading to wrong decisions in other cases.

However, for our auxiliary based approach (CLEF+Alpino_aux) with its regulated contribution of the general model, the results show that adding the feature does not help. On most datasets the same performance was achieved as by the in-domain model, while on only two datasets (CLEF 2005, 2007) the use of the auxiliary feature results in an insignificant improvement.

In contrast, simple model combination works surprisingly well. On two datasets (CLEF 2004 and 2005) this simple technique reaches a substantial improvement over *all* other models. On only one dataset (CLEF 2006) it falls slightly off the in-domain baseline, but still considerably outperforms data combination. This is true for both model combination methods, with estimated and equal weights. In general, the results show that model combination usually outperforms data combination (with the exception of one dataset, CLEF 2007), where, interestingly, the simplest model combination (equal weights) often performs best.

Contrary to expectations, the auxiliary based approach performs poorly and could often not even come close to the results obtained by simple model combination. In the following we will explore possible reasons for this result.

Examining possible causes One possible point of failure could be that the auxiliary feature was simply ignored. If the estimated weight would be close to zero the feature would indeed not contribute to the disambiguation task. Therefore, we examined the estimated weights for that feature. From that analysis we saw that, compared to the other features, the auxiliary feature got a weight relatively far from zero. It got on average a weight of -0.0905 in our datasets and as such is among the most influential weights, suggesting it to be important for disambiguation.

Another question that needs to be asked, however, is whether the feature is modeling properly the original Alpino model. For this sanity check, we create a model that contains only the single auxiliary feature and no other features. The feature's weight is set to a constant negative value[4]. The resulting model's performance is assessed on the complete CLEF data. The results (0% column in table 3) show that the auxiliary feature is indeed properly modeling the general Alpino model, as the two result in identical performance.

4.3.1 Feature template class models

In the experiments so far the general model was 'packed' into a single feature value. To check whether the feature alone is too weak, we examine the inclusion of several auxiliary distributions ($k > 1$). Each auxiliary feature we add represents a 'submodel' corresponding to an actual feature template class used in the original model. The feature's value is the negative log-probability as defined in equation 11, where OUT corresponds to the respective Alpino submodel.

The current Disambiguation Model of Alpino uses the 21 feature templates (van Noord and Malouf, 2005). Out of this given feature templates, we create two models that vary in the number of classes used. In the first model ('5 class'), we create five ($k = 5$) auxiliary distributions corresponding to five clusters of feature templates. They are

[4] Alternatively, we may estimate its weight, but as it does not have competing features we are safe to assume it constant.

defined manually and correspond to submodels for Part-of-Speech, dependencies, grammar rule applications, bilexical preferences and the remaining Alpino features. In the second model ('21 class'), we simply take every single feature template as its own cluster ($k = 21$).

We test the two models and compare them to our baseline. The results of this experiment are given in table 2. We see that both the 5 class and the 21 class model do not achieve any considerable improvement over the baseline (CLEF), nor over the single auxiliary model (CLEF+Alpino_aux).

Dataset (#sents)	5class	21class	CLEF+Alpino_aux	CLEF
CLEF2003 (446)	97.01	97.04	97.01	97.01
CLEF2004 (700)	96.57	96.60	96.60	96.60
CLEF2005 (200)	97.72	97.72	97.72	97.65
CLEF2006 (200)	97.06	97.06	97.06	97.06
CLEF2007 (200)	96.20	96.27	96.33	96.20

Table 2: Results on CLEF including several auxiliary features corresponding to Alpino submodels

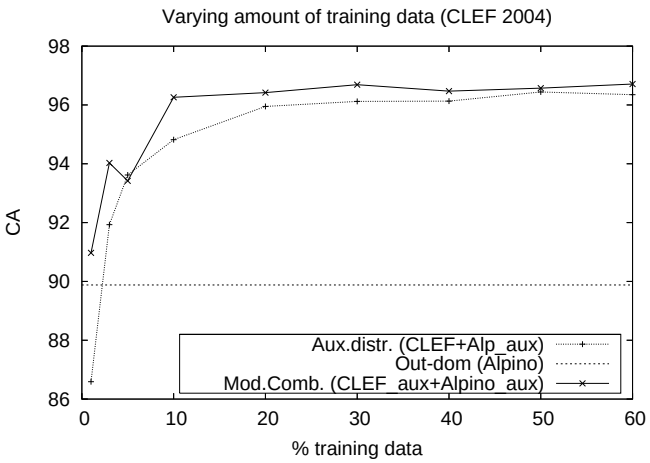

Figure 1: Amount of in-domain training data versus concept accuracy (Similar figures result from the other CLEF datasets) - note that we depict only aux.distr. as its performance is nearly indistinguishable from the in-domain (CLEF) baseline

4.3.2 Varying amount of training data

Our expectation is that the auxiliary feature is at least helpful in the case very little in-domain training data is available. Therefore, we evaluate the approach with smaller amounts of training data.

We sample (without replacement) a specific amount of training instances from the original QA data files and train models on the reduced training data. The resulting models are tested with and without the additional feature as well as model combination on the complete data set by using cross validation. Table 3 reports the results of these experiments for models trained on a proportion of up to 10% CLEF data. Figure 1 illustrates the overall change in performance.

Obviously, an increasing amount of in-domain training data improves the accuracy of the models. However, for our auxiliary feature, the results in table 3 show that the models with and without the auxiliary feature result in an overall almost identical performance (thus in figure 1 we depict only one of the lines). Hence, the inclusion of the auxiliary feature does not help in this case either. The models achieve similar performance even independently of the available amount of in-domain training data.

Thus, even on models trained on very little in-domain training data (e.g. 1% CLEF training data) the auxiliary based approach does not work. It even hurts performance, i.e. depending on the specific dataset, the inclusion of the auxiliary feature results in a model whose performance lies even *below* the original Alpino model accuracy, for up to a certain percentage of training data (varying on the dataset from 1% up to 10%).

In contrast, simple model combination is much more beneficial. It is able to outperform almost constantly the in-domain baseline (CLEF) and our auxiliary based approach (CLEF+Alpino_aux). Furthermore, in contrast to the auxiliary based approach, model combination never falls below the out-of-domain (Alpino) baseline, not even in the case a tiny amount of training data is available. This is true for both model combinations (estimated versus equal weights).

We would have expected the auxiliary feature to be useful at least when very little in-domain training data is available. However, the empirical results reveal the contrary[5]. We believe the reason for this drop in performance is the amount of available in-domain training data and the corresponding scaling of the auxiliary feature's weight. When little training data is available, the weight cannot be estimated reliably and hence is not contributing enough compared to the other features (exemplified in the drop of performance from 0% to 1%

[5] As suspected by a reviewer, the (non-auxiliary) features may overwhelm the single auxiliary feature, such that possible improvements by increasing the feature space on such a small scale might be invisible. We believe this is not the case. Other studies have shown that including just a few features might indeed help (Johnson and Riezler, 2000; van Noord, 2007). (e.g., the former just added 3 features).

	0%		1%				5%				10%			
Dataset	no aux	= Alp.	no aux	+aux	m.c.	eq.w.	no aux	+aux	m.c.	eq.w.	no aux	+aux	m.c.	eq.w.
CLEF2003	94.02	94.02	91.93	91.93	95.59	93.65	93.83	93.83	95.74	95.17	94.80	94.77	95.72	95.72
CLEF2004	89.88	89.88	86.59	86.59	90.97	91.06	93.62	93.62	93.42	92.95	94.79	94.82	96.26	95.85
CLEF2005	87.98	87.98	87.34	87.41	91.35	89.15	95.90	95.90	97.92	97.52	96.31	96.37	98.19	97.25
CLEF2006	88.92	88.92	89.64	89.64	92.16	91.17	92.77	92.77	94.98	94.55	95.04	95.04	95.04	95.47
CLEF2007	92.48	92.48	91.07	91.13	95.44	93.32	94.60	94.60	95.63	95.69	94.21	94.21	95.95	95.43

Table 3: Results on the CLEF data with varying amount of training data

training data in table 3). In such cases it is more beneficial to just apply the original Alpino model or the simple model combination technique.

4.4 Experiments with CGN

One might argue that the question domain is rather 'easy', given the already high baseline performance and the fact that few hand-annotated questions are enough to obtain a reasonable model. Therefore, we examine our approach on CGN (Oostdijk, 2000).

The empirical results of testing using cross-validation within a subset of CGN subdomains are given in table 4. The baseline accuracies are much lower on this more heterogeneous, spoken, data, leaving more room for potential improvements over the in-domain model. However, the results show that the auxiliary based approach does not work on the CGN subdomains either. The approach is not able to improve even on datasets where very little training data is available (e.g. comp-l), thus confirming our previous finding. Moreover, in some cases the auxiliary feature rather, although only slightly, *degrades* performance (indicated in italic in table 4) and performs worse than the counterpart model without the additional feature.

Depending on the different characteristics of data/domain and its size, the best model adaptation method varies on CGN. On some subdomains simple model combination performs best, while on others it is more beneficial to just apply the original, out-of-domain Alpino model.

To conclude, model combination achieves in most cases a modest improvement, while we have shown empirically that our domain adaptation method based on auxiliary distributions performs just similar to a model trained on in-domain data.

5 Conclusions

We examined auxiliary distributions (Johnson and Riezler, 2000) for domain adaptation. While the auxiliary approach has been successfully applied to lexical selectional preferences (Johnson and Riezler, 2000; van Noord, 2007), our empirical results show that integrating a more general into a domain-specific model through the auxiliary feature approach does not help. The auxiliary approach needs training data to estimate the weight(s) of the auxiliary feature(s). When little training data is available, the weight cannot be estimated appropriately and hence is not contributing enough compared to the other features. This result was confirmed on both examined domains. We conclude that the auxiliary feature approach is not appropriate for integrating information of a more general model to leverage limited in-domain data. Better results were achieved either without adaptation or by simple model combination.

Future work will consist in investigating other possibilities for parser adaptation, especially *semi-supervised* domain adaptation, where no labeled in-domain data is available.

References

Abney, Steven P. 1997. Stochastic attribute-value grammars. *Computational Linguistics*, 23:597–618.

Berger, A. and H. Printz. 1998. A comparison of criteria for maximum entropy / minimum divergence feature selection. In *In Proceedings of the 3nd Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 97–106, Granada, Spain.

Berger, Adam, Stephen Della Pietra, and Vincent Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–72.

Boros, M., W. Eckert, F. Gallwitz, G. Görz, G. Hanrieder, and H. Niemann. 1996. Towards understanding spontaneous speech: Word accuracy vs. concept accuracy. In *Proceedings of the Fourth International Conference on Spoken Language Processing (ICSLP 96)*, Philadelphia.

Briscoe, Ted, John Carroll, Jonathan Graham, and Ann Copestake. 2002. Relational evaluation schemes. In *Proceedings of the Beyond PARSEVAL Workshop at the 3rd International Conference on Language Resources and Evaluation*, pages 4–8, Las Palmas, Gran Canaria.

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

Hara, Tadayoshi, Miyao Yusuke, and Jun'ichi Tsujii. 2005. Adapting a probabilistic disambiguation model of an hpsg

comp-a (1,193) - Spontaneous conversations ('face-to-face')						comp-b (525) - Interviews with teachers of Dutch					
DataSet	no aux	+ aux	Alpino	Mod.Comb.	Mod.Comb. eq.weights	Dataset	no aux	+ aux	Alpino	Mod.Comb.	Mod.Comb eq.weights
fn000250	63.20	63.28	62.90	63.91	**63.99**	fn000081	66.20	66.39	66.45	**67.26**	66.85
fn000252	64.74	64.74	64.06	64.87	**64.96**	fn000089	62.41	62.41	63.88	**64.35**	64.01
fn000254	66.03	*66.00*	65.78	66.39	**66.44**	fn000086	62.60	62.76	63.17	63.59	**63.77**
comp-l (116) - Commentaries/columns/reviews (broadcast)						comp-m (267) - Ceremonious speeches/sermons					
DataSet	no aux	+ aux	Alpino	Mod.Comb.	Model.Comb. eq.weights	Dataset	no aux	+ aux	Alpino	Mod.Comb.	Mod.Comb eq.weights
fn000002	67.63	67.63	**77.30**	76.96	72.40	fn000271	59.25	59.25	63.78	**64.94**	61.76
fn000017	64.51	*64.33*	**66.42**	66.30	65.74	fn000298	70.33	*70.19*	74.55	**74.83**	72.70
fn000021	61.54	61.54	**64.30**	64.10	63.24	fn000781	72.26	72.37	**73.55**	**73.55**	73.04

Table 4: Excerpt of results on various CGN subdomains (# of sentences in parenthesis).

parser to a new domain. In *Proceedings of the International Joint Conference on Natural Language Processing*.

Johnson, Mark and Stefan Riezler. 2000. Exploiting auxiliary distributions in stochastic unification-based grammars. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 154–161, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Johnson, Mark, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic "unification-based" grammars. In *Proceedings of the 37th Annual Meeting of the ACL*.

Lease, Matthew, Eugene Charniak, Mark Johnson, and David McClosky. 2006. A look at parsing and its applications. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*, Boston, Massachusetts, 16–20 July.

McClosky, David, 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, USA, June. Association for Computational Linguistics.

Oostdijk, Nelleke. 2000. The Spoken Dutch Corpus: Overview and first evaluation. In *Proceedings of Second International Conference on Language Resources and Evaluation (LREC)*, pages 887–894.

Osborne, Miles. 2000. Estimation of stochastic attribute-value grammars using an informative sample. In *Proceedings of the Eighteenth International Conference on Computational Linguistics (COLING 2000)*.

Ratnaparkhi, A. 1997. A simple introduction to maximum entropy models for natural language processing. Technical report, Institute for Research in Cognitive Science, University of Pennsylvania.

van Noord, Gertjan and Robert Malouf. 2005. Wide coverage parsing with stochastic attribute value grammars. Draft available from http://www.let.rug.nl/˜vannoord. A preliminary version of this paper was published in the Proceedings of the IJCNLP workshop Beyond Shallow Analyses, Hainan China, 2004.

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*, pages 20–42, Leuven.

van Noord, Gertjan. 2007. Using self-trained bilexical preferences to improve disambiguation accuracy. In *Proceedings of the Tenth International Conference on Parsing Technologies. IWPT 2007, Prague.*, pages 1–10, Prague.

van Noord, Gertjan. In preparation. Learning efficient parsing.

Velldal, E. and S. Oepen. 2005. Maximum entropy models for realization ranking. In *Proceedings of MT-Summit*, Phuket, Thailand.