Semi-supervised Training of a Statistical Parser from Unlabeled Partially-bracketed Data

Rebecca Watson and Ted Briscoe
Computer Laboratory
University of Cambridge, UK
FirstName.LastName@cl.cam.ac.uk

John Carroll
Department of Informatics
University of Sussex, UK
J.A.Carroll@sussex.ac.uk

Abstract

We compare the accuracy of a statistical parse ranking model trained from a fully-annotated portion of the Susanne treebank with one trained from unlabeled partially-bracketed sentences derived from this treebank and from the Penn Treebank. We demonstrate that *confidence-based* semi-supervised techniques similar to self-training outperform expectation maximization when both are constrained by partial bracketing. Both methods based on partially-bracketed training data outperform the fully supervised technique, and both can, in principle, be applied to any statistical parser whose output is consistent with such partial-bracketing. We also explore tuning the model to a different domain and the effect of in-domain data in the semi-supervised training processes.

1 Introduction

Extant statistical parsers require extensive and detailed treebanks, as many of their lexical and structural parameters are estimated in a fully-supervised fashion from treebank derivations. Collins (1999) is a detailed exposition of one such ongoing line of research which utilizes the Wall Street Journal (WSJ) sections of the Penn Treebank (PTB). However, there are disadvantages to this approach. Firstly, treebanks are expensive to create manually. Secondly, the richer the annotation required, the harder it is to adapt the treebank to train parsers which make different assumptions about the structure of syntactic analyses. For example, Hockenmeier (2003) trains a statistical parser based on Combinatory Categorial Grammar (CCG) on the WSJ PTB, but first maps the treebank to CCG derivations semi-automatically. Thirdly, many (lexical) parameter estimates do not generalize well between domains. For instance, Gildea (2001) reports that WSJ-derived bilexical parameters in Collins' (1999) Model 1 parser contribute about 1% to parse selection accuracy when test data is in the same domain, but yield no improvement for test data selected from the Brown Corpus. Tadayoshi *et al.* (2005) adapt a statistical parser trained on the WSJ PTB to the biomedical domain by retraining on the Genia Corpus, augmented with manually corrected derivations in the same format. To make statistical parsing more viable for a range of applications, we need to make more effective and flexible use of extant training data and minimize the cost of annotation for new data created to tune a system to a new domain.

Unsupervised methods for training parsers have been relatively unsuccessful to date, including expectation maximization (EM) such as the inside-outside algorithm (IOA) over PCFGs (Baker, 1979; Prescher, 2001). However, Pereira and Schabes (1992) adapted the IOA to apply over semi-supervised data (unlabeled bracketings) extracted from the PTB. They constrain the training data (parses) considered within the IOA to those consistent with the constituent boundaries defined by the bracketing. One advantage of this approach is that, although less information is derived from the treebank, it gen-

Proceedings of the 10th Conference on Parsing Technologies, pages 23–32,
Prague, Czech Republic, June 2007.

eralizes better to parsers which make different representational assumptions, and it is easier, as Pereira and Schabes did, to map unlabeled bracketings to a format more consistent with the target grammar. Another is that the cost of annotation with unlabeled brackets should be lower than that of developing a representationally richer treebank. More recently, both Riezler *et al.* (2002) and Clark and Curran (2004) have successfully trained maximum entropy parsing models utilizing all derivations in the model consistent with the annotation of the WSJ PTB, weighting counts by the normalized probability of the associated derivation. In this paper, we extend this line of investigation by utilizing only unlabeled and partial bracketing.

We compare the performance of a statistical parsing model trained from a detailed treebank with that of the same model trained with semi-supervised techniques that require only unlabeled partially-bracketed data. We contrast an IOA-based EM method for training a PGLR parser (Inui *et al.*, 1997), similar to the method applied by Pereira and Schabes to PCFGs, to a range of *confidence-based* semi-supervised methods described below. The IOA is a generalization of the Baum-Welch or Forward-Backward algorithm, another instance of EM, which can be used to train Hidden Markov Models (HMMs). Elworthy (1994) and Merialdo (1994) demonstrated that Baum-Welch does not necessarily improve the performance of an HMM part-of-speech tagger when deployed in an unsupervised or semi-supervised setting. These somewhat negative results, in contrast to those of Pereira and Schabes (1992), suggest that EM techniques require fairly determinate training data to yield useful models. Another motivation to explore alternative non-iterative methods is that the derivation space over partially-bracketed data can remain large (>1K) while the *confidence-based* methods we explore have a total processing overhead equivalent to one iteration of an IOA-based EM algorithm.

As we utilize an initial model to annotate additional training data, our methods are closely related to self-training methods described in the literature (e.g. McClosky *et al.* 2006, Bacchiani *et al.* 2006). However these methods have been applied to fully-annotated training data to create the initial model, which is then used to annotate further training data derived from unannotated text. Instead, we train entirely from partially-bracketed data, starting from the small proportion of 'unambiguous' data whereby a single parse is consistent with the annotation. Therefore, our methods are better described as semi-supervised and the main focus of this work is the flexible re-use of existing treebanks to train a wider variety of statistical parsing models. While many statistical parsers extract a context-free grammar in parallel with a statistical parse selection model, we demonstrate that existing treebanks can be utilized to train parsers that deploy grammars that make other representational assumptions. As a result, our methods can be applied by a range of parsers to minimize the manual effort required to train a parser or adapt to a new domain.

§2 gives details of the parsing system that are relevant to this work. §3 and §4 describe our data and evaluation schemes, respectively. §5 describes our semi-supervised training methods. §6 explores the problem of tuning a parser to a new domain. Finally, §7 gives conclusions and future work.

2 The Parsing System

Sentences are automatically preprocessed in a series of modular pipelined steps, including tokenization, part of speech (POS) tagging, and morphological analysis, before being passed to the statistical parser. The parser utilizes a manually written feature-based unification grammar over POS tag sequences.

2.1 The Parse Selection Model

A context-free 'backbone' is automatically derived from the unification grammar[1] and a generalized or non-deterministic LALR(1) table is

[1] This backbone is determined by compiling out the values of prespecified attributes. For example, if we compile out the attribute PLURAL which has 2 possible values (plural or not) we will create 2 CFG rules for each rule with categories that contain PLURAL. Therefore, no information is lost during this process.

constructed from this backbone (Tomita, 1987). The residue of features not incorporated into the backbone are unified on each reduce action and if unification fails the associated derivation paths also fail. The parser creates a packed parse forest represented as a graph-structured stack.[2] The parse selection model ranks complete derivations in the parse forest by computing the product of the probabilities of the (shift/reduce) parse actions (given LR state and lookahead item) which created each derivation (Inui *et al.*, 1997).

Estimating action probabilities, consists of a) recording an action history for the correct derivation in the parse forest (for each sentence in a treebank), b) computing the frequency of each action over all action histories and c) normalizing these frequencies to determine probability distributions over conflicting (i.e. shift/reduce or reduce/reduce) actions.

Inui *et al.* (1997) describe the probability model utilized in the system where a transition is represented by the probability of moving from one stack state, σ_{i-1}, (an instance of the graph structured stack) to another, σ_i. They estimate this probability using the stack-top state s_{i-1}, next input symbol l_i and next action a_i. This probability is conditioned on the type of state s_{i-1}. S_s and S_r are mutually exclusive sets of states which represent those states reached after shift or reduce actions, respectively. The probability of an action is estimated as:

$$P(l_i, a_i, \sigma_i|\sigma_{i-1}) \approx \left\{ \begin{array}{ll} P(l_i, a_i|s_{i-1}) & s_{i-1} \in S_s \\ P(a_i|s_{i-1}, l_i) & s_{i-1} \in S_r \end{array} \right\}$$

Therefore, normalization is performed over all lookaheads for a state or over each lookahead for the state depending on whether the state is a member of S_s or S_r, respectively (hereafter the I function). In addition, Laplace estimation can be used to ensure that all actions in the table are assigned a non-zero probability (the I_L function).

3 Training Data

The treebanks we use in this work are in one of two possible formats. In either case, a treebank T consists of a set of sentences. Each sentence t is a pair (s, M), where s is the automatically preprocessed set of POS tagged tokens (see §2) and M is either a fully annotated derivation, A, or an unlabeled bracketing U. This bracketing may be partial in the sense that it may be compatible with more than one derivation produced by a given parser. Although occasionally the bracketing is itself complete but alternative nonterminal labeling causes indeterminacy, most often the 'flatter' bracketing available from extant treebanks is compatible with several alternative 'deeper' mostly binary-branching derivations output by a parser.

3.1 Derivation Consistency

Given $t = (s, A)$, there will exist a single derivation in the parse forest that is compatible (correct). In this case, equality between the derivation tree and the treebank annotation A identifies the correct derivation. Following Pereira and Schabes (1992) given $t = (s, U)$, a node's span in the parse forest is *valid* if it does not overlap with any span outlined in U, and hence, a derivation is correct if the span of every node in the derivation is valid in U. That is, if no crossing brackets are present in the derivation. Thus, given $t = (s, U)$, there will often be more than one derivation compatible with the partial bracketing.

Given the correct nodes in the parse forest or in derivations, we can then extract the corresponding action histories and estimate action probabilities as described in §2.1. In this way, partial bracketing is used to constrain the set of derivations considered in training to those that are compatible with this bracketing.

3.2 The Susanne Treebank and Baseline Training Data

The Susanne Treebank (Sampson, 1995) is utilized to create fully annotated training data.

[2] The parse forest is an instance of a *feature forest* as defined by Miyao and Tsujii (2002). We will use the term 'node' herein to refer to an element in a derivation tree or in the parse forest that corresponds to a (sub-)analysis whose label is the mother's label in the corresponding CF 'backbone' rule.

This treebank contains detailed syntactic derivations represented as trees, but the node labeling is incompatible with the system grammar. We extracted sentences from Susanne and automatically preprocessed them. A few multiwords are retokenized, and the sentences are retagged using the POS tagger, and the bracketing deterministically modified to more closely match that of the grammar, resulting in a bracketed corpus of 6674 sentences. We will refer to this bracketed treebank as S, henceforth.

A fully-annotated and system compatible treebank of 3543 sentences from S was also created. We will refer to this annotated treebank, used for fully supervised training, as B. The system parser was applied to construct a parse forest of analyses which are compatible with the bracketing. For 1258 sentences, the grammar writer interactively selected correct (sub)analyses within this set until a single analysis remained. The remaining 2285 sentences were automatically parsed and all consistent derivations were returned. Since B contains more than one possible derivation for roughly two thirds of the data the 1258 sentences (paired with a single tree) were repeated twice so that counts from these trees were weighted more highly. The level of reweighting was determined experimentally using some held out data from S. The baseline supervised model against which we compare in this work is defined by the function $I_L(B)$ as described in §2.1. The costs of deriving the fully-annotated treebank are high as interactive manual disambiguation takes an average of ten minutes per sentence, even given the partial bracketing derived from Susanne.

3.3 The WSJ PTB Training Data

The Wall Street Journal (WSJ) sections of the Penn Treebank (PTB) are employed as both training and test data by many researchers in the field of statistical parsing. The annotated corpus implicitly defines a grammar by providing a labeled bracketing over words annotated with POS tags. We extracted the unlabeled bracketing from the de facto standard training sections (2-21 inclusive).[3] We will refer to the resulting corpus as W and the combination (concatenation) of the partially-bracketed corpora S and W as SW.

3.4 The DepBank Test Data

King *et al.* (2003) describe the development of the PARC 700 Dependency Bank, a gold-standard set of relational dependencies for 700 sentences (from the PTB) drawn at random from section 23 of the WSJ (the de facto standard test set for statistical parsing). In all the evaluations reported in this paper we test our parser over a gold-standard set of relational dependencies compatible with our parser output derived (Briscoe and Carroll, 2006) from the PARC 700 Dependency Bank (DepBank, henceforth).

The Susanne Corpus is a (balanced) subset of the Brown Corpus which consists of 15 broad categories of American English texts. All but one category (reportage text) is drawn from different domains than the WSJ. We therefore, following Gildea (2001) and others, consider S, and also the baseline training data, B, as out-of-domain training data.

4 The Evaluation Scheme

The parser's output is evaluated using a relational dependency evaluation scheme (Carroll, *et al.*, 1998; Lin, 1998) with standard measures: precision, recall and F_1. Relations are organized into a hierarchy with the root node specifying an unlabeled dependency. The microaveraged precision, recall and F_1 scores are calculated from the counts for all relations in the hierarchy which subsume the parser output. The microaveraged F_1 score for the baseline system using this evaluation scheme is 75.61%, which – over similar sets of relational dependencies – is broadly comparable to recent evaluation results published by King and collaborators with their state-of-the-art parsing system (Briscoe *et al.*, 2006).

[3] The pipeline is the same as that used for creating S though we do not automatically map the bracketing to be more consistent with the system grammar, instead, we simply removed unary brackets.

4.1 Wilcoxon Signed Ranks Test

The Wilcoxon Signed Ranks (Wilcoxon, henceforth) test is a *non-parametric* test for statistical significance that is appropriate when there is one data sample and several measures. For example, to compare the accuracy of two parsers over the same data set. As the number of samples (sentences) is large we use the normal approximation for z. Siegel and Castellan (1988) describe and motivate this test. We use a 0.05 level of significance, and provide z-value probabilities for significant results reported below. These results are computed over microaveraged F_1 scores for each sentence in DepBank.

5 Training from Unlabeled Bracketings

We parsed all the bracketed training data using the baseline model to obtain up to 1K top-ranked derivations and found that a significant proportion of the sentences of the potential set available for training had only a single derivation compatible with their unlabeled bracketing. We refer to these sets as the *unambiguous training data* (γ) and will refer to the remaining sentences (for which more than one derivation was compatible with their unlabeled bracketing) as the *ambiguous training data* (α). The availability of significant quantities of unambiguous training data that can be found automatically suggests that we may be able to dispense with the costly reannotation step required to generate the fully supervised training corpus, B.

Table 1 illustrates the split of the corpora into mutually exclusive sets γ, α, 'no match' and 'timeout'. The latter two sets are not utilized during training and refer to sentences for which all parses were inconsistent with the bracketing and those for which no parses were found due to time and memory limitations (self-imposed) on the system.[4] As our grammar is different from that implicit in the WSJ PTB there is a high proportion of sentences where no parses were consistent with the unmodified PTB bracketing. However, a preliminary investigation of no matches didn't yield any clear patterns of inconsistency that we could quickly ameliorate by simple modifications of the PTB bracketing. We leave for the future a more extensive investigation of these cases which, in principle, would allow us to make more use of this training data. An alternative approach that we have also explored is to utilize a similar bootstrapping approach with data partially-annotated for grammatical relations (Watson and Briscoe, 2007).

Corpus	$\mid \gamma \mid$	$\mid \alpha \mid$	No Match	Timeout
S	1097	4138	1322	191
W	6334	15152	15749	1094
SW	7409	19248	16946	1475

Table 1: Corpus split for S, W and SW.

5.1 Confidence-Based Approaches

We use γ to build an initial model. We then utilize this initial model to derive derivations (compatible with the unlabeled partial bracketing) for α from which we select additional training data. We employ two types of selection methods. First, we select the top-ranked derivation only and weight actions which resulted in this derivation equally with those of the initial model (C_1). This method is similar to 'Viterbi training' of HMMs though we do not weight the corresponding actions using the top parse's probability. Secondly, we select more than one derivation, placing an appropriate weight on the corresponding action histories based on the initial model's confidence in the derivation. We consider three such models, in which we weight transitions corresponding to each derivation ranked r with probability p in the set of size n either using $\frac{1}{n}$, $\frac{1}{r}$ or p itself to weight counts.[5] For example, given a treebank T with sentences $t = (s, U)$, function P to return the set of parses consistent with U given t and function A that returns the set of actions given a parse p, then the frequency count of action a_k in C_r is

[4] As there are time and memory restrictions during parsing, the SW results are not equal to the sum of those from S and W analysis.

[5] In §2.1 we calculate action probabilities based on frequency counts where we perform a weighted sum over action histories and each history has a weight of 1. We extend this scheme to include weights that differ between action histories corresponding to each derivation.

determined as follows:

$$| a_k |= \sum_{i=1}^{|T|} \sum_{j=1,a_k \in A(p_{ij})}^{|P(t_i)|} \frac{1}{j}$$

These methods all perform normalization over the resulting action histories using the training function I_L and will be referred to as C_n, C_r and C_p, respectively. C_n is a 'uniform' model which weights counts only by degree of ambiguity and makes no use of ranking information. C_r weights counts by derivation rank, and C_p is simpler than and different to one iteration of EM as outside probabilities are not utilized. All of the semi-supervised functions described here take two arguments: an initial model and the data to train over, respectively.

Models derived from unambiguous training data, γ, alone are relatively accurate, achieving indistinguishable performance to that of the baseline system given either W or SW as training data. We utilize these models as initial models and train over different corpora with each of the confidence-based models. Table 2 gives results for all models. Results statistically significant compared to the baseline system are shown in bold print (better) or italic print (worse). These methods show promise, often yielding systems whose performance is significantly better than the baseline system. Method C_r achieved the best performance in this experiment and remained consistently better in those reported below. Throughout the different approaches a domain effect can be seen, models utilizing just S are worse, although the best performing models benefit from the use of both S and W as training data (i.e. SW).

Model	Prec	Rec	F$_1$	$P(z)^‡$
Baseline	77.05	74.22	75.61	-
$I_L(\gamma(S))$	76.02	73.40	*74.69*	0.0294
$C_1(I_L(\gamma(S)), \alpha(S))$	77.05	74.22	75.61	0.4960
$C_n(I_L(\gamma(S)), \alpha(S))$	77.51	74.80	76.13	0.0655
$C_r(I_L(\gamma(S)), \alpha(S))$	77.73	74.98	**76.33**	0.0154
$C_p(I_L(\gamma(S)), \alpha(S))$	76.45	73.91	75.16	0.2090
$I_L(\gamma(W))$	77.01	74.31	75.64	0.1038
$C_1(I_L(\gamma(W)), \alpha(W))$	76.90	74.23	75.55	0.2546
$C_n(I_L(\gamma(W)), \alpha(W))$	77.85	75.07	**76.43**	0.0017
$C_r(I_L(\gamma(W)), \alpha(W))$	77.88	75.04	**76.43**	0.0011
$C_p(I_L(\gamma(W)), \alpha(W))$	77.40	74.75	76.05	0.1335
$I_L(\gamma(SW))$	77.09	74.35	75.70	0.1003
$C_1(I_L(\gamma(SW)), \alpha(SW))$	76.86	74.21	75.51	0.2483
$C_n(I_L(\gamma(SW)), \alpha(SW))$	77.88	75.05	**76.44**	0.0048
$C_r(I_L(\gamma(SW)), \alpha(SW))$	78.01	75.13	**76.54**	0.0007
$C_p(I_L(\gamma(SW)), \alpha(SW))$	77.54	74.95	76.23	0.0618

Table 2: Performance of all models on DepBank. ‡represents the statistical significance of the system against the baseline model.

5.2 EM

Our EM model differs from that of Pereira and Schabes as a PGLR parser adds *context* over a PCFG so that a single rule can be applied in several different states containing reduce actions. Therefore, the summation and normalization performed for a CFG rule within IOA is instead applied within such contexts. We can apply I (our PGLR normalization function without Laplace smoothing) to perform the required steps if we output the action history with the corresponding normalized inside-outside weight for each node (Watson *et al.*, 2005).

We perform EM starting from two initial models; either a uniform probability model, $I_L()$, or from models derived from unambiguous training data, γ. Figure 1 shows the cross entropy decreasing monotonically from iteration 2 (as guaranteed by the EM method) for different corpora and initial models. Some models show an initial increase in cross-entropy from iteration 1 to iteration 2, because the models are initialized from a subset of the data which is used to perform maximisation. Cross-entropy increases, by definition, as we incorporate ambiguous data with more than one consistent derivation.

Performance over DepBank can be seen in Figures 2, 3, and 4 for each dataset S, W and SW, respectively. Comparing the C_r and EM lines in each of Figures 2, 3, and 4, it is evident that C_r outperforms EM across all datasets, regardless of the initial model applied. In most cases, these results are significant, even when we manually select the best model (iteration) for EM.

The graphs of EM performance from iteration 1 illustrate the same 'classical' and 'initial' patterns observed by Elworthy (1994). When EM is initialized from a relatively poor model, such as that built from S (Figure 2), a 'classical'

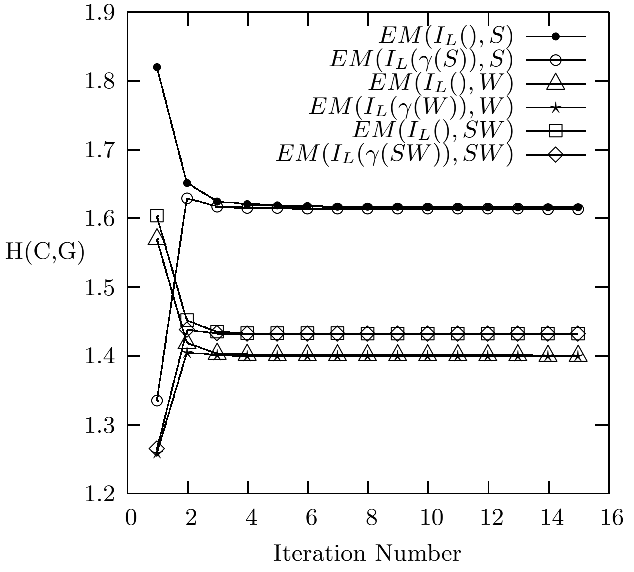

Figure 1: Cross Entropy Convergence for various training data and models, with EM.

pattern emerges with relatively steady improvement from iteration 1 until performance asymptotes. However, when the starting point is better (Figures 3 and 4), the 'initial' pattern emerges in which the best performance is reached after a single iteration.

6 Tuning to a New Domain

When building NLP applications we would want to be able to tune a parser to a new domain with minimal manual effort. To obtain training data in a new domain, annotating a corpus with partial-bracketing information is much cheaper than full annotation. To investigate whether such data would be of value, we considered W to be the corpus over which we were tuning and applied the best performing model trained over S, $C_r(I_L(\gamma(S)), \alpha(S))$, as our initial model. Figure 5 illustrates the performance of C_r compared to EM.

Tuning using C_r was not significantly different from the model built directly from the entire data set with C_r, achieving 76.57% as opposed to 76.54% F_1 (see Table 2). By contrast, EM performs better given all the data from the beginning rather than tuning to the new domain.

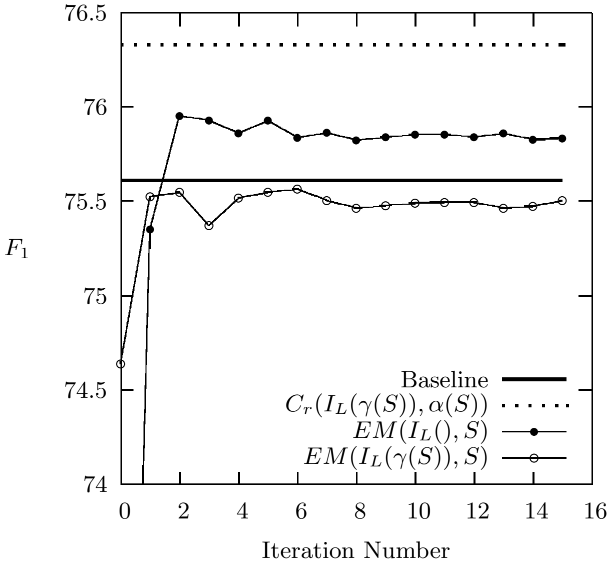

Figure 2: Performance over S for C_r and EM.

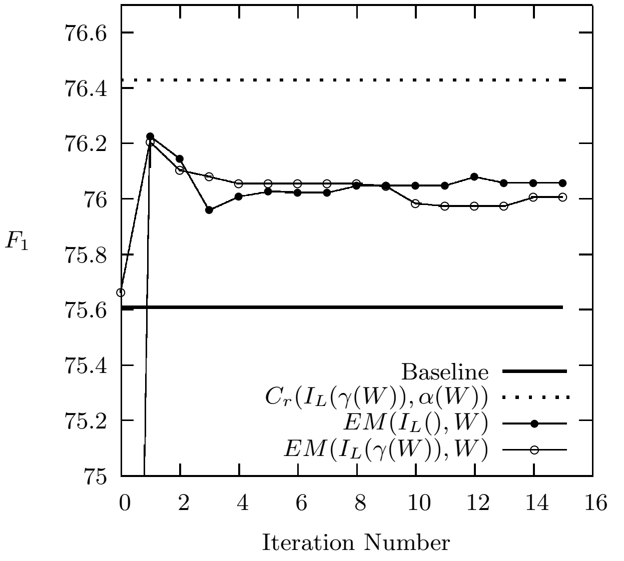

Figure 3: Performance over W for C_r and EM.

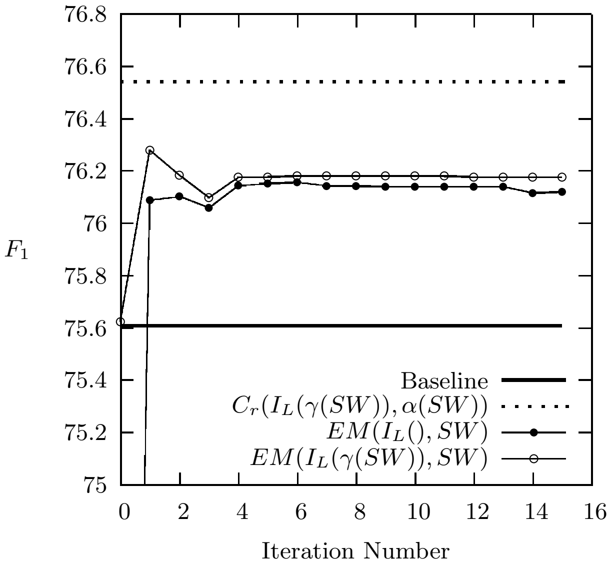

Figure 4: Performance over SW for C_r and EM.

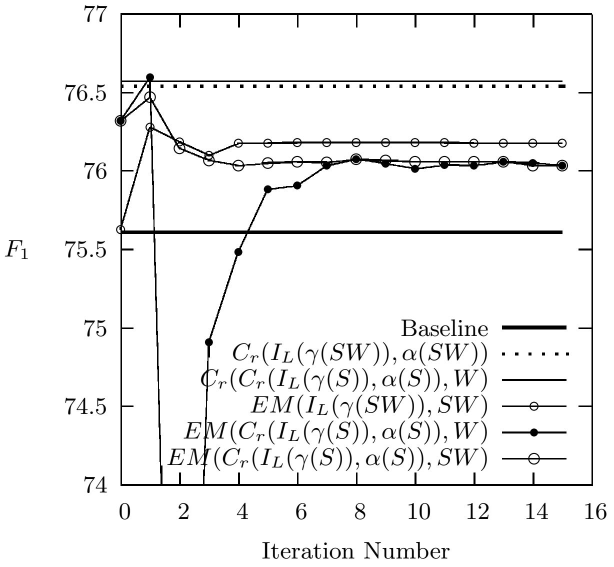

Figure 5: Tuning over the WSJ PTB (W) from Susanne Corpus (S).

C_r generally outperforms EM, though it is worth noting the behavior of EM given only the tuning data (W) rather than the data from both domains (SW). In this case, the graph illustrates a combination of Elworthy's 'initial' and 'classical' patterns. The steep drop in performance (down to 69.93% F_1) after the first iteration is probably due to loss of information from S. However, this run also eventually converges to similar performance, suggesting that the information in S is effectively disregarded as it forms only a small portion of SW, and that these runs effectively converge to a local maximum over W.

Bacchiani *et al.* (2006), working in a similar framework, explore weighting the contribution (frequency counts) of the in-domain and out-of-domain training datasets and demonstrate that this can have beneficial effects. Furthermore, they also tried unsupervised tuning to the in-domain corpus by weighting parses for it by their normalized probability. This method is similar to our C_p method. However, when we tried unsupervised tuning using the WSJ and an initial model built from S in conjunction with our confidence-based methods, performance degraded significantly.

7 Conclusions

We have presented several semi-supervised *confidence-based* training methods which have significantly improved performance over an extant (more supervised) method, while also reducing the manual effort required to create training or tuning data. We have shown that given a medium-sized unlabeled partially bracketed corpus, the confidence-based models achieve superior results to those achieved with EM applied to the same PGLR parse selection model. Indeed, a bracketed corpus provides flexibility as existing treebanks can be utilized despite the incompatibility between the system grammar and the underlying grammar of the treebank. Mapping an incompatible annotated treebank to a compatible partially-bracketed corpus is relatively easy compared to mapping to a compatible fully-annotated corpus.

An immediate benefit of this work is that (re)training parsers with incrementally-modified grammars based on different linguistic frameworks should be much more straightforward – see, for example Oepen *et al.* (2002) for a good discussion of the problem. Furthermore, it suggests that it may be possible to usefully tune a parser to a new domain with less annotation effort.

Our findings support those of Elworthy (1994) and Merialdo (1994) for POS tagging and suggest that EM is not always the most suitable semi-supervised training method (especially when some in-domain training data is available). The confidence-based methods were successful because the level of noise introduced did not outweigh the benefit of incorporating all derivations compatible with the bracketing in which the derivations contained a high proportion of correct constituents. These findings may not hold if the level of bracketing available does not adequately constrain the parses considered – see Hwa (1999) for a related investigation with EM.

In future work we intend to further investigate the problem of tuning to a new domain, given that minimal manual effort is a major priority. We hope to develop methods which required no manual annotation, for example, high precision automatic partial bracketing using phrase chunking and/or named entity recognition techniques might yield enough information to support the training methods developed here.

Finally, further experiments on weighting the contribution of each dataset might be beneficial. For instance, Bacchiani *et al.* (2006) demonstrate imrpovements in parsing accuracy with unsupervised adaptation from unannotated data and explore the effect of different weighting of counts derived from the supervised and unsupervised data.

Acknowledgements

The first author is funded by the Overseas Research Students Awards Scheme, and the Poynton Scholarship awarded by the Cambridge Australia Trust in collaboration with the Cambridge Commonwealth Trust. Development of the RASP system was and is supported by the EPSRC (grants GR/N36462, GR/N36493 and GR/T19919).

References

Bacchiani, M., Riley, M., Roark, B. and R. Sproat (2006) 'MAP adaptation of stochastic grammars', *Computer Speech and Language, vol.20.1,* pp.41–68.

Baker, J. K. (1979) 'Trainable grammars for speech recognition' in Klatt, D. and Wolf, J. (eds.), *Speech Communications Papers for the 97th Meeting of the Acoustical Society of America,* MIT, Cambridge, Massachusetts, pp. 557–550.

Briscoe, E.J., J. Carroll and R. Watson (2006) 'The Second Release of the RASP System', *Proceedings of ACL-Coling'06,* Sydney, Australia.

Carroll, J., Briscoe, T. and Sanfilippo, A. (1998) 'Parser evaluation: a survey and a new proposal', *Proceedings of LREC,* Granada, pp. 447–454.

Clark, S. and J. Curran (2004) 'Parsing the WSJ Using CCG and Log-Linear Models', *Proceedings of 42nd Meeting of the Association for Computational Linguistics,* Barcelona, pp. 103–110.

Collins, M. (1999) *Head-driven Statistical Models for Natural Language Parsing,* PhD Dissertation, University of Pennsylvania.

Elworthy, D. (1994) 'Does Baum-Welch Reestimation Help Taggers?', *Proceedings of ANLP,* Stuttgart, Germany, pp. 53–58.

Gildea, D. (2001) 'Corpus variation and parser performance', *Proceedings of EMNLP,* Pittsburgh, PA.

Hockenmaier, J. (2003) *Data and models for statistical parsing with Combinatory Categorial Grammar,* PhD Dissertation, The University of Edinburgh.

Hwa, R. (1999) 'Supervised grammar induction using training data with limited constituent information', *Proceedings of ACL,* College Park, Maryland, pp. 73–79.

Inui, K., V. Sornlertlamvanich, H. Tanaka and T. Tokunaga (1997) 'A new formalization of probabilistic GLR parsing', *Proceedings* *of IWPT,* MIT, Cambridge, Massachusetts, pp. 123–134.

King, T.H., R. Crouch, S. Riezler, M. Dalrymple and R. Kaplan (2003) 'The PARC700 Dependency Bank', *Proceedings of LINC,* Budapest.

Lin, D. (1998) 'Dependency-based evaluation of MINIPAR', *Proceedings of Workshop at LREC'98 on The Evaluation of Parsing Systems,* Granada, Spain.

McClosky, D., Charniak, E. and M. Johnson (2006) 'Effective self-training for parsing', *Proceedings of HLT-NAACL,* New York.

Merialdo, B. (1994) 'Tagging English Text with a Probabilistic Model', *Computational Linguistics, vol.20.2,* pp.155–171.

Miyao, Y. and J. Tsujii (2002) 'Maximum Entropy Estimation for Feature Forests', *Proceedings of HLT,* San Diego, California.

Oepen, S., K. Toutanova, S. Shieber, C. Manning, D. Flickinger, and T. Brants (2002) 'The LinGO Redwoods Treebank: Motivation and preliminary applications', *Proceedings of COLING,* Taipei, Taiwan.

Pereira, F and Y. Schabes (1992) 'Inside-Outside Reestimation From Partially Bracketed Corpora', *Proceedings of ACL,* Delaware.

Prescher, D. (2001) 'Inside-outside estimation meets dynamic EM', *Proceedings of 7th Int. Workshop on Parsing Technologies (IWPT01),* Beijing, China.

Riezler, S., T. King, R. Kaplan, R. Crouch, J. Maxwell III and M. Johnson (2002) 'Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques', *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics,* Philadelphia, pp. 271–278.

Sampson, G. (1995) *English for the Computer,* Oxford University Press, Oxford, UK.

Siegel S. and N. J. Castellan (1988) *Nonparametric Statistics for the Behavioural Sciences, 2nd edition,* McGraw-Hill.

Tadayoshi, H., Y. Miyao and J. Tsujii (2005) 'Adapting a probabilistic disambiguation model of an HPSG parser to a new domain', *Proceedings of IJCNLP,* Jeju Island, Korea.

Tomita, M. (1987) 'An Efficient Augmented Context-Free Parsing Algorithm', *Computational Linguistics, vol.13(1–2),* pp.31–46.

Watson, R. and E.J. Briscoe (2007) 'Adapting the RASP system for the CoNLL07 domain-adaptation task', *Proceedings of EMNLP-CoNLL-07,* Prague.

Watson, R., J. Carroll and E.J. Briscoe (2005) 'Efficient extraction of grammatical relations', *Proceedings of 9th Int. Workshop on Parsing Technologies (IWPT'05),* Vancouver, Ca..