MAP Adaptation of Stochastic Grammars

Michiel Bacchiani [a] Michael Riley [b] Brian Roark [a]
and Richard Sproat [c]

[a] *AT&T Labs-Research, 180 Park Ave., Florham Park, NJ 07932, USA*
{michiel,roark}@research.att.com

[b] *Google Inc., 1440 Broadway, New York, NY 10018, USA*
riley@google.com

[c] *Departments of Linguistics and ECE, University of Illinois at Urbana-Champaign*
Foreign Languages Building 4103, 707 South Mathews Avenue, MC-168
Urbana, IL, 61801
rws@uiuc.edu

Abstract

This paper investigates supervised and unsupervised adaptation of stochastic grammars, including n-gram language models and probabilistic context-free grammars (PCFGs), to a new domain. It is shown that the commonly used approaches of count merging and model interpolation are special cases of a more general Maximum *a posteriori* (MAP) framework, which additionally allows for alternate adaptation approaches. This paper investigates the effectiveness of different adaptation strategies, and, in particular, focuses on the need for supervision in the adaptation process. We show that n-gram models as well as PCFGs benefit from either supervised or unsupervised MAP adaptation in various tasks. For n-gram models, we compare the benefit from supervised adaptation with that of unsupervised adaptation on a speech recognition task with an adaptation sample of limited size (about 17 hours), and show that unsupervised adaptation can obtain 51% of the 7.7% adaptation gain obtained by supervised adaptation. We also investigate the benefit of using multiple word hypotheses (in the form of a word lattice) for unsupervised adaptation on a speech recognition task for which there was a much larger adaptation sample available. The use of word lattices for adaptation required the derivation of a generalization of the well-known Good-Turing estimate. Using this generalization, we derive a method that uses Monte Carlo sampling for building Katz backoff models. The adaptation results show that, for adaptation samples of limited size (several tens of hours), unsupervised adaptation on lattices gives a performance gain over using transcripts. The experimental results also show that with a very large adaptation sample (1050 hours), the benefit from transcript based adaptation matches that of lattice based adaptation. Finally, we show that PCFG domain adaptation using the MAP framework provides similar gains in F-measure accuracy on a parsing task as was seen in ASR accuracy improvements with n-gram adaptation. Experimental results show that unsupervised adaptation provides 37% of the 10.35% gain

obtained by supervised adaptation.

Key words: A, B, C
PACS: 001

1 Introduction

Most current speech and language processing systems rely on statistical modeling, requiring large quantities of annotated training data for parameter estimation of the system. The performance of the system on test data depends in large part on how well the statistical characteristics of the training material match that of the test data. Developing a system for a new domain, which will have its own statistical characteristics, is costly, primarily due to the collection and preparation of the training data. In particular, manual annotation of training data is very labor intensive.

In an attempt to decrease the cost of developing systems for new domains, domain adaptation has received a fair amount of attention. Such approaches try to leverage *out-of-domain* models or data (models or data mismatched to the domain of interest) to decrease the *in-domain* annotation requirements. This requires an algorithm to derive an adapted model from out-of-domain data or from an out-of-domain model, when given a sample of in-domain data, that is possibly small and might or might not be annotated.

In acoustic modeling for Automatic Speech Recognition (ASR), adaptation based on a small data sample has received a large amount of attention. For large vocabulary systems, an effective acoustic model will have millions of free parameters; observations for only a small subset of the parameters will be available in the adaptation sample. Hence direct re-estimation will be plagued by data sparsity and possibly futile, since only a small fraction of the model will be affected. One class of algorithms addresses that problem by use of affine transformations applied to all distributions in the model [23,9]. A second class of algorithms approaches acoustic model adaptation by estimating those distributions for which there were observations in the adaptation sample, and handles data sparsity by smoothing the estimate based on the adaptation sample with the previous model estimate. Most notably in this class of algorithms is the maximum *a posteriori* (MAP) adaptation algorithm [10] which considers the model parameter estimates themselves a random variable with a given prior distribution. The adapted parameter estimates are found as the mode of the posterior distribution obtained from the prior and unadapted model distributions. Since this technique only adapts those distributions that were seen in the adaptation sample, it does not perform as well as transform

based approaches when the adaptation sample is very small (e.g. less than 60 seconds of speech) but will outperform the transform based techniques with larger adaptation samples, as it is less constrained.

In contrast to adaptation of acoustic models for ASR, adaptation of stochastic grammars, such as n-gram models used for language modeling or probabilistic context-free grammars (PCFGs) for statistical parsing, has received much less attention. The most widespread approaches to n-gram adaptation in a large vocabulary setting are model interpolation (e.g. [36]) and count mixing (e.g. [24]). The domain adaptation of statistical parsing models described in [11] is essentially the same as the count mixing approach commonly use in n-gram adaptation.

Another area of focus to reduce the development effort of building a system for a new domain is the use of unsupervised or lightly supervised learning. There it is assumed that the effort of collecting data from a domain of interest falls short of full manual annotation. Out-of-domain information is leveraged either by automatic annotation using an out-of-domain trained system, or by using an automatic selection algorithm (active learning) to sample the in-domain data and then manually annotate that (hopefully most informative) subset. In lightly supervised training it is assumed that some annotation is available for the in-domain sample, but that this annotation is noisy.

In the field of acoustic modeling for ASR, [22] showed that it is possible to obtain accurate acoustic models in a lightly supervised setup, using as little as 10 minutes of supervised training data. The manually annotated sample is used to bootstrap the acoustic model of the system and a language model is trained on the noisy transcripts. That system is then used to automatically transcribe the in-domain data and those annotations are used to re-estimate the model.

Unsupervised n-gram adaptation for ASR has also been investigated recently in [14,33]. [33] used the unweighted transcripts to build language models; [14] filtered or weighted based on confidence measures. The confidence annotation in that work was obtained from consensus hypothesis decoding [25].

For statistical parsing models, [16] demonstrated how active learning techniques can reduce the amount of annotated data required to converge on the best performance, by selecting from among the candidate strings to be annotated in ways which promote more informative examples for earlier annotation. [15] used a variant of the inside-outside algorithm presented in [26] to exploit a partially labeled out-of-domain treebank, and found an advantage to adaptation over direct grammar induction.

Another direction of generalization that arises when studying n-gram adaptation based on unsupervised annotated data is that recognizers can produce

multiple hypotheses in the form of weighted word lattices. These lattices provide a probability distribution over a number of competing hypothesis transcriptions, and hence contain more information that could potentially be exploited for adaptation.

The focus of this paper is on adaptation of stochastic grammars, in particular n-gram models and PCFGs [1]. Using the derivation of the MAP algorithm developed for acoustic model adaptation in ASR, we show in section 2 that most adaptation approaches previously used for n-grams and PCFGs are special cases of the more general MAP framework. In section 3 we compare n-gram adaptation with and without supervision. Particular attention is paid to using unsupervised adaptation with multiple hypotheses, i.e. using lattices rather than transcripts. In section 4 we compare the use of supervised and unsupervised adaptation for a PCFG-based statistical parser.

2 Maximum a Posteriori-based Adaptation

In the MAP estimation framework described in detail in [10], the model parameters θ are assumed to be a random vector in the space Θ. Given an observation sample $\mathbf{x}$, the MAP estimate is obtained as the mode of the posterior distribution of θ denoted as $g(. \mid \mathbf{x})$

$$\theta_{\text{MAP}} = \underset{\theta}{\operatorname{argmax}}\, g(\theta \mid \mathbf{x}) = \underset{\theta}{\operatorname{argmax}}\, f(\mathbf{x} \mid \theta) g(\theta) \tag{1}$$

Although the derivation in [10] was aimed at estimation of Gaussian mixture distributions, it generalizes directly for use in n-gram and PCFG adaptation. In an n-gram language model, model states represent word-histories and a multinomial distribution is defined for the possible words following that history. In a PCFG, a state represents a set of production rules with a multinomial distribution across those rules. This is entirely analogous to the distribution across mixture components within a mixture density. Instead of estimating mixture weights, here the task is to estimate word or rule probabilities. Following the motivation and derivation in [10], a practical candidate for the prior distribution of the weights $\omega_1, \omega_2, \cdots, \omega_K$ is the Dirichlet density,

$$g(\omega_1, \omega_2, \cdots, \omega_K \mid \nu_1, \nu_2, \cdots, \nu_K) \propto \prod_{i=1}^{K} \omega_i^{\nu_i - 1} \tag{2}$$

where $\nu_i > 0$ are the parameters of the Dirichlet distribution. If the expected counts for state s are denoted as C_s and for the i-th component as $c_{s,i}$, the

[1] Some of the results reported here were first reported in [2], [32] and [28].

mode of the posterior distribution is obtained as

$$\hat{\omega}_{s,i} = \frac{(\nu_{s,i} - 1) + c_{s,i}}{\sum_{k=1}^{K}(\nu_{s,k} - 1) + C_s} \quad 1 \leq i \leq K. \tag{3}$$

Note that there are as many free parameters in the prior distribution as there are in the model, hence this in itself makes the approach not practical for adaptation purposes where one wants to control the number of free parameters for the sake of robustness with sparse data. Generally, a tying or parameterization of the prior distributions is used to limit the number of free parameters in the prior distribution. The frequently used model interpolation and count merging approaches correspond to two choices of parameterizations of the prior distributions in this general MAP framework.

We will define the expected counts $c^{\mathbf{w}}_{s,i}$ from a sample $\mathbf{w}$ of size $|\mathbf{w}|$, for use in equation 3, as follows. Let $\mathrm{P}^{\mathbf{w}}(s, \omega_i)$ be the joint probability of s and the i-th component, estimated from the sample $\mathbf{w}$ in a manner which may reserve probability mass for unobserved events. Then $c^{\mathbf{w}}_{s,i} = |\mathbf{w}|\,\mathrm{P}^{\mathbf{w}}(s, \omega_i)$ and $C^{\mathbf{w}}_s = \sum_i c^{\mathbf{w}}_{s,i}$. We will denote the raw count as $\hat{c}^{\mathbf{w}}_{s,i}$.

Let expected counts and probability estimates from the out-of-domain data or model be denoted with superscript ^{O} and their in-domain counterparts with superscript ^{I}. Then a count merging approach with mixing parameters α and β is obtained by choosing the parameters of the prior distribution as

$$\nu_{s,i} = C^O_s \frac{\alpha}{\beta} \mathrm{P}^O(\omega_i \mid s) + 1 \tag{4}$$

since in that case

$$\begin{aligned} \widehat{\mathrm{P}}_{mrg}(\omega_i \mid s) &= \frac{C^O_s \frac{\alpha}{\beta} \mathrm{P}^O(\omega_i \mid s) + c^I_{s,i}}{\sum_{k=1}^{K} \left[C^O_s \frac{\alpha}{\beta} \mathrm{P}^O(\omega_k \mid s) \right] + C^I_s} \\ &= \frac{\alpha c^O_{s,i} + \beta c^I_{s,i}}{\alpha C^O_s + \beta C^I_s}. \end{aligned} \tag{5}$$

On the other hand, if the parameters of the prior distribution are chosen as

$$\nu_{s,i} = C^I_s \frac{\lambda}{1 - \lambda} \mathrm{P}^O(\omega_i \mid s) + 1 \tag{6}$$

the MAP estimate reduces to a model interpolation approach with parameter λ, since in that case

$$\begin{aligned}\widehat{\mathrm{P}}_{intp}(\omega_i \mid s) &= \frac{C_s^I \frac{\lambda}{1-\lambda}\mathrm{P}^O(\omega_i \mid s) + c_{s,i}^I}{\sum_{k=1}^K \left[C_s^I \frac{\lambda}{1-\lambda}\mathrm{P}^O(\omega_k \mid s)\right] + C_s^I} \\ &= \frac{\frac{\lambda}{1-\lambda}\mathrm{P}^O(\omega_i \mid s) + \mathrm{P}^I(\omega_i \mid s)}{\frac{\lambda}{1-\lambda} + 1} \\ &= \lambda \mathrm{P}^O(\omega_i \mid s) + (1-\lambda)\mathrm{P}^I(\omega_i \mid s). \end{aligned} \tag{7}$$

The MAP framework unifies these two approaches as particular choices for the parameterization of the prior distribution. It also opens the possibility for other adaptation approaches by changing the prior parameterization.

3 MAP Adaptation of n-gram models

In this section, the MAP adaptation framework is applied to n-gram models. The MAP formulation directly applies to this type of model, with the states s in equation 3 corresponding to word histories (consisting of zero or more previous words) and the weights $\omega_{s,i}$ corresponding to the probability estimates for words w_i emitted from the state. The objective is to use the adapted models for ASR and hence the performance of the adapted models is evaluated based on the transcription accuracy of the resulting system.

All of the n-gram models are smoothed using Katz backoff [20]. The smoothing of the adapted n-gram models falls out of the use of expected counts in equation 3, and smoothed probability estimates in equations 4 and 6. That is, the MAP estimation provides probability mass to unobserved events because P^O and P^I do. The model is then defined as

$$\mathrm{P}(\omega_i \mid s) = \begin{cases} \widehat{\mathrm{P}}(\omega_i \mid s) & \text{if } \hat{c}_{s,i}^I + \hat{c}_{s,i}^O > 0 \\ \alpha\, \mathrm{P}(\omega_i \mid s') & \text{otherwise} \end{cases} \tag{8}$$

where α is the backoff weight and s' the backoff history for history s.

First, in section 3.1 we investigate the effects of the parameterization of the prior distribution and compare the effectiveness of unsupervised adaptation with supervised adaptation. In addition we consider hybrid approaches, using supervised adaptation for a subset of the in-domain sample and unsupervised adaptation for the rest. In the unsupervised adaptation experiments, we limit our scope to the use of single transcripts rather than lattices representing multiple hypotheses. In addition, the unsupervised experiments are limited by the fairly small amount of available adaptation data (17 hours).

In section 3.2 we expand our scope to using unsupervised adaptation based on

either transcripts or lattices. We experiment in a different domain where we have a much larger adaptation sample available (over 1000 hours), allowing us to compare the benefit from using the lattice representation over using transcripts for different sizes of the adaptation sample. An important difference for unsupervised adaptation between single transcripts and probability weighted multiple transcripts obtained from a lattice is the fact that the resulting counts are integers in one case and fractional counts in the other. Fractional counts violate an assumption made in the commonly used Good-Turing estimation. As a result, section 3.2 describes in detail the generalization of Good-Turing estimation used to estimate the smoothing parameters.

3.1 Adaptation based on transcripts

To evaluate the effectiveness of the transcript-based adaptation, we measured the transcription accuracy of an adapted voicemail transcription system on voicemail messages received at a customer care line of a telecommunications network center. The initial voicemail system, named Scanmail, was trained on general voicemail messages collected from the mailboxes of people at our research site in Florham Park, NJ. The target domain is also composed of voicemail messages, but for a mailbox that receives messages from customer care agents regarding network outages. In contrast to the general voicemail messages from the training corpus of the Scanmail system, the messages from the target domain, named SSNIFR, are focused solely on network related problems. They contain frequent mention of various network related acronyms and trouble ticket numbers, rarely (if at all) found in the training corpus of the Scanmail system.

The transcription system used in these experiments is described in section 3.1.1. The experimental results obtained using various adaptation approaches are described in section 3.1.2 and discussed in section 3.1.3.

3.1.1 System Description

To evaluate the transcription accuracy, we used a multi-pass speech recognition system that employs various unsupervised speaker and channel normalization techniques. An initial search pass produces word-lattice output that is used as the grammar in subsequent search passes. The system is almost identical to the one described in detail in [1]. The main differences in terms of the acoustic model of the system are the use of linear discriminant analysis features; use of a 100 hour training set as opposed to a 60 hour training set; and the modeling of the speaker gender which in this system is identical to that described in [35]. Note that the acoustic model is appropriate for either domain as the

messages are collected on a voicemail system of the same type. This parallels the experiments in [22], where the focus was on AM adaptation in the case where the LM was deemed appropriate.

The language model of the Scanmail system is a Katz backoff trigram, trained on hand-transcribed messages of approximately 100 hours of voicemail (1 million words). The model contains 13460 unigram, 175777 bigram, and 495629 trigram probabilities. The lexicon of the Scanmail system contains 13460 words and was compiled from all the unique words found in the 100 hours of transcripts of the Scanmail training set.

3.1.2 Experimental Results

For every experiment, we report the accuracy of the one-best transcripts obtained at 4 stages of the recognition process, after the first pass lattice construction (denoted as FP), after vocal tract length normalization and gender modeling (denoted as VTLN), after Constrained Model-space Adaptation (denoted as CMA) and after Maximum Likelihood Linear regression adaptation (denoted as MLLR).

For the SSNIFR domain we have available a 1 hour manually transcribed test set (10819 words) and approximately 17 hours of manually transcribed adaptation data (163343 words). In all experiments, the vocabulary of the system is left unchanged. Generally, for a domain shift this can raise the error rate significantly due to an increase in the OOV rate. However, this increase in the experiments here is limited because the majority of the new domain-dependent vocabulary are acronyms which are covered by the Scanmail vocabulary through individual letters. The OOV rate of the SSNIFR test set, using the Scanmail vocabulary is 2%.

Table 1 lists the results obtained using 3.7 hours (38586 words) of manually transcribed SSNIFR domain data. The baseline result is the performance of the Scanmail system on the 1 hour SSNIFR test set without any adaptation. The in-domain result was obtained using a trigram language model trained on the 3.7 hours of in-domain data alone. The other rows give the performance of systems using the Scanmail language model, adapted with either count merging (equation 5) or interpolation (equation 7). It can be seen that both adaptation approaches improve performance over the baseline (28.0%) and also improve over the in-domain trained model (26.2%). There is a larger improvement for the count merge adaptation than for the interpolation adaptation (5.8% vs. 5.4%). The count merging parameters ($\alpha = 1$ and $\beta = 5$) and interpolation parameter ($\lambda = 0.75$) were obtained empirically. Given these results, all subsequent experiments used a count merging approach with the same merging parameters.

System	FP	VTLN	CMA	MLLR
Baseline	32.7	30.0	28.3	28.0
In-domain	29.4	27.3	26.5	26.2
Count Merging	26.3	23.4	22.6	22.2
Interpolation	26.6	23.7	23.0	22.6

Table 1
Recognition performance using 3.7 hours of in-domain data for either training or adaptation using count merging or interpolation. The merging parameters were $\alpha = 1$ and $\beta = 5$, the interpolation parameter was $\lambda = 0.75$.

Fraction of the adaptation set (%)	FP	VTLN	CMA	MLLR
0	32.7	30.0	28.3	28.0
25	25.6	23.2	22.3	22.0
50	24.8	21.8	21.3	21.1
75	23.8	21.6	20.8	20.4
100	23.7	21.1	20.5	20.3

Table 2
Recognition on the 1 hour SSNIFR test set using systems obtained by supervised LM adaptation on various sized subsets of the 17 hour adaptation set.

Table 2 shows the results from supervised adaptation of the Scanmail language model using different sized subsets of the 17 hours of SSNIFR adaptation material. In these experiments, LM adaptation counts are obtained from the manual transcripts rather than from ASR transcripts. Note that training directly on the 17 hours of SSNIFR adaptation data, without using the out-of-domain data, yields an MLLR-stage word-error rate of 22.8, which is 2.5 percent worse than using both the in-domain and out-of-domain data.

Table 3 repeats this experiment but in an unsupervised setting. Each subset of the adaptation data was first transcribed using an ASR system with the Scanmail language model. These transcripts were then used to obtain counts, and the Scanmail language model was adapted using those counts. Although most of the improvement in accuracy comes from adapting on just 25% of the available 17 hours, improvements in both FP and MLLR accuracy were had by increasing the size of the adaptation sample.

Both supervised and unsupervised LM adaptation give performance improvements over the baseline using no adaptation. On a quarter of the 17 hour adaptation set, the unsupervised LM adaptation gives a 2.5% drop in the word error rate, compared to a 6.0% reduction using supervised LM adap-

Fraction of the adaptation set (%)	FP	VTLN	CMA	MLLR
0	32.7	30.0	28.3	28.0
25	28.9	27.0	25.8	25.5
50	28.4	26.0	25.2	24.8
75	28.1	25.6	24.9	24.7
100	28.2	25.6	24.9	24.6

Table 3
Recognition on the 1 hour SSNIFR test set using systems obtained by unsupervised LM adaptation on various sized subsets of the 17 hour adaptation set.

tation. Increasing the amount of data used for LM adaptation to the full 17 hours gives an additional 1.7% and 0.9% improvement for the supervised and unsupervised cases respectively.

To investigate the effect of iterative LM adaptation, we used the system obtained by unsupervised LM adaptation on all of the 17 hour adaptation set to re-transcribe the entire adaptation set. We then used the counts from the MLLR-pass transcripts, together with the counts from the Scanmail language model, to obtain an adapted model. The results of adapted systems at multiple iterations are shown in table 4. A second iteration provided an additional 0.5% accuracy improvement. A third iteration gave no improvement in accuracy.

Iterations of adaptation	FP	VTLN	CMA	MLLR
0	32.7	30.0	28.3	28.0
1	28.2	25.6	24.9	24.6
2	27.9	25.1	24.4	24.1
3	28.0	25.3	24.7	24.3

Table 4
Recognition results of systems obtained by iterations of unsupervised LM adaptation using the entire 17 hour adaptation set. The adaptation counts were obtained from transcription with an adapted system.

To see to what extent the improvements of iterative LM adaptation are dependent on the starting point, we transcribed the adaptation set using the system obtained by supervised LM adaptation on 25% of the adaptation set. We then constructed adapted language models using the Scanmail model counts, the 25% supervised counts, and the counts obtained from the MLLR transcripts for the remaining subsets of the adaptation set. Both the supervised and unsupervised counts from the adaptation set were weighted with the same mixing

Fraction of the adaptation set (%)	FP	VTLN	CMA	MLLR
50	25.4	22.2	21.7	21.5
100	25.0	22.1	21.5	21.3

Table 5
Recognition results of systems obtained by a second iteration of unsupervised LM adaptation using various sized subsets of the 17 hour adaptation set. The 50% row consists of 25% supervised, 25% unsupervised; the 100% row consists of 25% supervised, 75% unsupervised. The automatic transcription for the unsupervised adaptation was done with a 25% supervised adapted system, hence the baseline is the 25% row of table 2.

parameters $\alpha = 1$ and $\beta = 5$. The possibility of using different parameters for the supervised and unsupervised counts was not investigated to allow a more direct comparison of the results of this mixed approach with that of the supervised-only results. However, given the difference in reliability of the supervised and unsupervised transcripts, it is possible that using multiple parameters can result in improved accuracy. The results of the system adapted on the mixed supervised and unsupervised counts are shown in table 5. A comparison of these results with the performance of the system obtained just with supervised LM adaptation (table 2) demonstrates that using MLLR transcript-based counts in addition to to the supervised counts provides an additional accuracy improvement (21.3% vs. 22.0%) over using the supervised counts alone.

An alternative to an adaptation approach is to use the unsupervised counts obtained from ASR transcripts for model training directly. Table 6 shows the result using language models built from the MLLR transcripts of the adaptation set obtained by the baseline system. Using half of the adaptation set in this manner gave a 2% improvement in first-pass accuracy over the baseline; but this improvement is not additive, yielding just 0.4% improvement after all of the AM adaptation. The results do improve with more adaptation data: 2.9% FP accuracy improvement and 1.7% MLLR accuracy improvement.

Fraction of the adaptation set (%)	FP	VTLN	CMA	MLLR
50	30.7	28.4	27.7	27.6
100	29.8	27.0	26.4	26.3

Table 6
Recognition results of systems obtained by training language models solely from the transcripts produced by the baseline system on various subsets of the adaptation set.

A final LM adaptation scenario that was investigated on this data set is based

Initial model	FP	VTLN	CMA	MLLR
Scanmail	27.3	27.0	26.6	26.7
SSNIFR 17h unsup	25.5	24.5	24.1	24.0

Table 7
Recognition results of systems obtained by self-adaptation on the test set. Adaptation counts were obtained from the MLLR-pass test set transcripts produced by a system using the Scanmail or second iteration unsupervised adapted (see table 4) language models.

on self-adaptation. In this scenario, the adaptation counts are obtained from the MLLR transcripts produced by the final search pass on the 1 hour test set. The test set is then re-transcribed using a language model obtained by adaptation using the Scanmail counts and the adaptation counts from the test set. Table 7 shows the results from two such experiments. The experiments differed in the language model used for self-adaptation. In each experiment, the LM to be adapted was used to transcribe the test set. This LM was then adapted with the counts from the ASR transcript of the test set. One experiment used the baseline Scanmail language model; the other used the language model obtained by two iterations of unsupervised adaptation on the 17 hour adaptation set (see table 4). In both experiments, there is a large gain in the first-pass (FP) accuracy: 5.4% for the Scanmail trial (27.3% vs. 32.7%); 2.4% for the unsupervised adapted trial (25.5% vs. 27.9%). These gains, however, are not additive with the AM adaptation gains and reduce at the final search pass to 1.3% for the Scanmail trial (26.7% vs. 28.0%) and 0.1% for the unsupervised adapted trial (24.0% vs. 24.1%). This shows that the self-adaptation incorporates a part of the unsupervised AM adaptation gain. Self-adaptation does provide a gain in accuracy, but dependent on the starting point, since transcription accuracy improved for the baseline trial but not for the unsupervised adapted trial. The 1.3% improvement using self-adaptation alone on the baseline model is less than the 3.4% obtained by a single iteration of unsupervised adaptation on the 17 hour adaptation set.

3.1.3 Conclusions

The experimental results show various approaches to supervised and unsupervised language model adaptation based on counts from manually annotated transcripts or ASR transcripts. All experiments showed improved transcription performance compared to unadapted baseline system. Starting from a 28% word-error baseline, using 17 hours of in-domain adaptation data, supervised adaptation gives a 7.7% gain (20.3% vs. 28.0%); unsupervised LM adaptation achieves 51% of that gain (24.1% vs. 28.0%). A quarter of the 17 hour adaptation set, in a unsupervised setting, provides a 2.5% gain over the baseline (25.5% vs. 28.0%).

Iterative LM adaptation also improves accuracy, raising the accuracy gain from 3.4% to 3.9% with one additional iteration of the unsupervised adaptation approach. When starting with a model obtained by supervised adaptation on 25% of the adaptation set, iterative unsupervised adaptation still provides an additional improvement, raising the 6.0% gain from supervised adaptation by 0.7%.

Comparing the iterative unsupervised adaptation approach to a training approach, it shows that for a 17 hour adaptation sample, the gain from adaptation is 2.2% larger than that of training (3.9% vs. 1.7%).

Furthermore, self-adaptation on the 1 hour test set provides gains over the baseline system of 1.3%. This gain is, however, dependent on the starting point, since self-adaptation applied on top of an adapted model did not provide any additional gains. All self-adaptation experiments show a large improvement in the first-pass accuracy, however, this gain is not additive with the gains obtained from the AM normalization and self-adaptation algorithms. The LM adaptation process obtains part of the AM adaptation benefits by adapting on the transcripts that already have merited from AM adaptation. All other adaptation scenarios, however, were additive with AM normalization and self-adaptation. As a result, for subsequent trials, we simply report first pass results.

3.2 Adaptation based on lattices

In this section, the focus is on adaptation from word-lattices as opposed to using ASR transcripts. Lattices provide a probability distribution over a number of competing hypothesis transcriptions and hence contain more information than ASR transcripts. Adapting on ASR transcripts has led to effective model adaptation, but the question remains whether it is possible to get further improvements by taking into account the distribution over transcriptions provided by the word lattice.

Perhaps the first idea one might have for how to use lattices in n-gram modeling would be to use the probability distribution on the lattices to define expected n-gram counts. Note, in general, this leads to fractional n-gram counts. Some smoothing techniques, such as deleted interpolation [17] and Witten-Bell smoothing [34] could straight-forwardly use these counts, since these techniques rely on mixing maximum likelihood (relative frequency) estimates. However, other techniques, such as Katz backoff [20] or Kneser-Ney estimation [21], which are very widely used, rely on integral counts in their definition; exactly how to generalize them is not immediately clear. For this reason, we take the time here to outline a general approach to extending arbi-

trary smoothing techniques to lattice corpora. Using Good-Turing smoothing, we then employ a Monte Carlo version of our generalization for Katz back-off modeling and compare to the now standard technique of using one-best transcription.

After presenting our generalization for use with word lattices in section 3.2.1, we provide empirical results for adaptation to a novel customer call classification application in section 3.2.2. The baseline ASR results in this domain are below 50 percent word accuracy, which would lead to the expectation that the word lattice would provide better adaptation to the new domain than the error-filled ASR transcript. We show that, although the bulk of the accuracy gain through adaptation is also achieved using just ASR one-best transcripts, there is a consistent benefit to sampling from word lattices, enough to make this an interesting area of future research. In particular, the word lattice sampling approach converges more quickly, leading to more than a half percentage point improvement over the one-best method with 50 hours of unlabeled data. The one-best approach ultimately reaches the same level of performance, with more unlabeled examples, both in terms of word accuracy and perplexity. Based on the results that we present here, one can conclude that using the one-best transcripts is generally an effective approximation to using the word lattices.

3.2.1 Word Probability Estimation

In this section, we develop how to estimate word probabilities when drawn from a hidden sample, which we will apply to language model estimation from ASR word lattices. We first begin with the standard, known sample case.

3.2.1.1 Known random sample Let $\mathbf{w} = w_1 w_2 \dots w_N$ be a random sample of size N from a finite set $\mathcal{W}$ of words having probability distribution $\mathrm{P}(w)$. We assume that the distribution $\mathrm{P}(w)$ is unknown to us and we wish to estimate it. The maximum likelihood estimate is

$$\widehat{\mathrm{P}}_{ml}(w|\mathbf{w}) = \frac{r}{N}, \tag{9}$$

where $r = c(w, \mathbf{w})$ is the number of occurrences of w in the sample $\mathbf{w}$. However, it is relatively poor for low count words in the sample [12].

Good's estimate, which improves the estimate for low counts, is

$$\widehat{\mathrm{P}}_{g}(w|\mathbf{w}) = \frac{r+1}{N+1}\frac{\mathcal{E}_{N+1}[n_{r+1}]}{\mathcal{E}_{N}[n_r]}, \tag{10}$$

where n_r is the number of distinct words that have count r in a sample [12]

and $\mathcal{E}_N$ denotes the expectation over a sample of size N. Good shows that his estimate for the probability of a word w is equal to $\mathcal{E}[\mathrm{P}_u | c(u, \mathbf{w}) = c(w, \mathbf{w})]$, the expected value of the probability of a word u selected equiprobably from among those words with the same count as w in the sample.

The population quantity, $\mathcal{E}_N[n_r]$, in Good's estimate is unlikely to be known. In that case, it must be approximated. Turing's estimate

$$\widehat{\mathrm{P}}_t(w|\mathbf{w}) = \frac{r+1}{N}\frac{n_{r+1}}{n_r} \tag{11}$$

approximates $\mathcal{E}_N[n_r]$ with n_r from the sample [12].

Katz shows how to apply the Good-Turing estimate not just to words but to n-grams of words for building a stochastic LM [20].

3.2.1.2 Hidden random sample Consider now that the sample is also unknown to us. Instead we are only given $\mathcal{L} = (\mathcal{H}, \mathrm{P}(\mathbf{w}|\mathcal{L}))$, where $\mathcal{H} \subset \mathcal{W}^N$ is the set of sample hypotheses and $\mathrm{P}(\mathbf{w}|\mathcal{L}) > 0$ is the probability that $\mathbf{w} \in \mathcal{H}$ was indeed the sample. When $\mathbf{w} \in \mathcal{W}^N - \mathcal{H}$, we define $\mathrm{P}(\mathbf{w}|\mathcal{L}) = 0$. This models situations where we have imperfect knowledge about a sample. In ASR, for example, $\mathcal{L}$ might be the set of hypotheses and the corresponding (recognizer estimate of the) probabilities of their being correct for a passage of N words of speech. [2]

We wish to estimate $\mathrm{P}(w)$ from $\mathcal{L}$. Define

$$\widetilde{\mathrm{P}}(w|\mathcal{L}) = \sum_{\mathbf{w}\in\mathcal{H}} \widehat{\mathrm{P}}(w|\mathbf{w})\,\mathrm{P}(\mathbf{w}|\mathcal{L}), \tag{12}$$

where $\widehat{\mathrm{P}}$ is whatever estimator we have chosen to use when the sample is known. In other words, $\widetilde{\mathrm{P}}$ is the expected value of $\widehat{\mathrm{P}}$ given our sample hypotheses and their probabilities.

We choose $\widetilde{\mathrm{P}}$ as our estimator of $\mathrm{P}(w)$ from $\mathcal{L}$ since we believe equation 12 makes it a natural candidate and because this estimator has several desirable properties that we list in the next section. Before this, we need to consider $\widetilde{\mathrm{P}}$ under some restrictions on $\widehat{\mathrm{P}}$.

Suppose that $\widehat{\mathrm{P}}(w|\mathbf{w})$ only depends on the count r of w in $\mathbf{w}$, i.e. $r = c(w, \mathbf{w})$.

[2] The assumption that each hypothesis has fixed length N simplifies the analysis in this section. A plausible hypothesis from a recognizer of a long passage would be close but not always identical in length to what was, in fact, uttered.

This is true for the maximum likelihood and Good's estimate. Then

$$\widetilde{\mathrm{P}}(w|\mathcal{L}) = \sum_{\mathbf{w}\in\mathcal{H}} \widehat{\mathrm{P}}(w|r)\mathrm{P}(\mathbf{w}|\mathcal{L}) = \sum_{\mathbf{r}=0}^{N} \widehat{\mathrm{P}}(w|r)\mathrm{P}(r|w,\mathcal{L}) \tag{13}$$

thus, in these cases we can determine $\widetilde{\mathrm{P}}$ from $\mathrm{P}(r|w,\mathcal{L})$.

For the maximum likelihood estimator, equation 13 further simplifies to

$$\widetilde{\mathrm{P}}(w|\mathcal{L}) = \frac{1}{N}\sum_{\mathbf{r}=0}^{N} r\,\mathrm{P}(r|w,\mathcal{L}) = \frac{\mathcal{E}[r|w,\mathcal{L}]}{N} \equiv \frac{\bar{r}}{N}. \tag{14}$$

For Good's estimate, we need an estimate of $\mathcal{E}_N[n_r]$ to use equation 13. We can use

$$\mathcal{E}_N[n_r] = \sum_{w\in\mathcal{W}} \mathrm{P}_N(r|w) \;\approx \sum_{w\in\mathcal{W}} \mathrm{P}(r|w,\mathcal{L}). \tag{15}$$

3.2.1.3 Properties of $\widetilde{\mathrm{P}}$: Our estimator $\widetilde{\mathrm{P}}$ defined in equation 12 has the following properties:

(1) **$\widetilde{\mathrm{P}}$ reduces to $\widehat{\mathrm{P}}$ when $|\mathcal{H}| = 1$:**
If $\mathcal{H} = \{\mathbf{w_0}\}$ then $\widetilde{\mathrm{P}}(w|\mathcal{L}) = \widehat{\mathrm{P}}(w|\mathbf{w}_0)$. In the ASR example, this would correspond to perfect recognition of $\mathbf{w_0}$ and the two estimates coincide.

(2) **Bias**$(\widetilde{\mathrm{P}})$ = **Bias**$(\widehat{\mathrm{P}})$:

$$\begin{aligned}
\mathcal{E}[\widetilde{\mathrm{P}}(w)] &= \sum_{\mathcal{L}} \widetilde{\mathrm{P}}(w|\mathcal{L})\,\mathrm{P}(\mathcal{L}) \\
&= \sum_{\mathcal{L}}\sum_{\mathbf{w}\in\mathcal{H}} \widehat{\mathrm{P}}(w|\mathbf{w})\,\mathrm{P}(\mathbf{w}|\mathcal{L})\,\mathrm{P}(\mathcal{L}) \\
&= \sum_{\mathbf{w}\in\mathcal{W}^N} \widehat{\mathrm{P}}(w|\mathbf{w}) \sum_{\mathcal{L}} \mathrm{P}(\mathbf{w}|\mathcal{L})\,\mathrm{P}(\mathcal{L}) \\
&= \sum_{\mathbf{w}\in\mathcal{W}^N} \widehat{\mathrm{P}}(w|\mathbf{w})\,\mathrm{P}(\mathbf{w}) \\
&= \mathcal{E}[\widehat{\mathrm{P}}(w)]
\end{aligned} \tag{16}$$

so $\widetilde{\mathrm{P}}$ has the same bias as $\widehat{\mathrm{P}}$ and, in particular, is unbiased if $\widehat{\mathrm{P}}$ is.

(3) **Consistency of $\widetilde{\mathrm{P}}$:**
Let $\mathbf{w}_0$ be the hidden sample of size N. If $\mathrm{P}_N(r_0|w,\mathcal{L})$ converges in probability to 1 as $N \to \infty$, where $r_0 = c(w,\mathbf{w_0})$, and if $\widehat{\mathrm{P}}(w|r)$ is weakly consistent, then $\widetilde{\mathrm{P}}(w|\mathcal{L})$ is weakly consistent. This follows since if $|\widehat{\mathrm{P}}(w|r_0) - \mathrm{P}(w)| < \epsilon$ and $1 - \mathrm{P}_N(r_0|w,\mathcal{L}) < \epsilon$, then

$$
\begin{aligned}
|\widetilde{\mathrm{P}}(w|\mathcal{L}) - \mathrm{P}(w)| &= \left| \sum_{r=0}^{N} \widehat{\mathrm{P}}(w|r)\ \mathrm{P}_N(r|w,\mathcal{L}) - \mathrm{P}(w) \right| \\
&\le |\widehat{\mathrm{P}}(w|r_0)\ \mathrm{P}_N(r_0|w,\mathcal{L}) - \mathrm{P}(w)| + \\
&\quad \sum_{\substack{r=0 \\ r \neq r_0}}^{N} \widehat{\mathrm{P}}(w|r)\ \mathrm{P}_N(r|w,\mathcal{L}) \\
&\le |\widehat{\mathrm{P}}(w|r_0)\ \mathrm{P}_N(r_0|w,\mathcal{L}) - \mathrm{P}(w)| + \\
&\quad 1 - \mathrm{P}_N(r_0|w,\mathcal{L}) \\
&\le 3\epsilon. \qquad (17)
\end{aligned}
$$

(4) **Expected value of the probability of same count words:**
Consider the expected value of the probability of a word u, selected equiprobably from among those words with the same count as w in a sample $\mathbf{w}$. When the sample is known, this equals Good's estimate, $\widehat{\mathrm{P}}_g$. When $\mathbf{w}$ is hidden, it is

$$
\begin{aligned}
\mathcal{E}[\mathrm{P}_u|c(u,\mathbf{w}) = c(w,\mathbf{w})] &= \sum_{\mathbf{w}\in\mathcal{H}} \mathcal{E}[\mathrm{P}_u|c(u,\mathbf{w}) = c(w,\mathbf{w}),\mathbf{w}]\ \mathrm{P}(\mathbf{w}|\mathcal{L}) \\
&= \sum_{r=0}^{N} \mathcal{E}[\mathrm{P}_u|r]\ \mathrm{P}(r|w,\mathcal{L}) \\
&= \sum_{r=0}^{N} \widehat{\mathrm{P}}_g(w|r)\ \mathrm{P}(r|w,\mathcal{L}) \qquad (18)
\end{aligned}
$$

and we see from equation 13 that equation 18 equals $\widetilde{\mathrm{P}}$ when $\widehat{\mathrm{P}}$ is Good's estimate. Thus our generalization of Good's estimate when the sample is hidden equals the expected value of the probability of a word selected equiprobably from among those with the same count as w just as when the sample is known.

3.2.1.4 Computing $\widetilde{\mathbf{P}}$ We can compute the $\widetilde{\mathrm{P}}$ in equation 12 approximately by using Monte Carlo methods. For this, we first generate M random samples, $\mathbf{w}_1, \ldots, \mathbf{w}_M \in \mathcal{H}$, from $\mathrm{P}(\mathbf{w}|\mathcal{L})$ and approximate

$$
\widetilde{\mathrm{P}}(w|\mathcal{L}) \approx \frac{1}{M} \sum_{i=1}^{M} \widehat{\mathrm{P}}(w|\mathbf{w}_i). \qquad (19)
$$

For more direct methods, we need to restrict $\widehat{\mathrm{P}}$. Let us require that it depends only on the count r as in equation 13. In order to use equation 13, we need to evaluate

$$
\mathrm{P}(r|w,\mathcal{L}) = \sum_{\substack{\mathbf{w}\in\mathcal{H}, \\ c(w,\mathbf{w})=r}} \mathrm{P}(\mathbf{w}|\mathcal{L}). \qquad (20)
$$

This, in principle, can be computed by enumerating all word sequences in $\mathcal{H}$ and explicitly forming the sum in the equation. However, this may not be practical when $|\mathcal{H}|$ is large. In that case, we may be able to divide and conquer. For suppose $\mathcal{L}$ can be divided into two independent parts $\mathcal{L}_1$ and $\mathcal{L}_2$ where for $i \in \{1, 2\}$, $\mathcal{L}_i = (\mathcal{H}_i, \mathrm{P}(\mathbf{w_i}, \mathcal{L}_i))$, $\mathcal{H}_i \subset \mathcal{W}^{N_i}$, $N_1 + N_2 = N$, $\mathcal{H} = \mathcal{H}_1\mathcal{H}_2$, and $\mathrm{P}(\mathbf{w_1 w_2}|\mathcal{L}) = \mathrm{P}(\mathbf{w_1}|\mathcal{L}_2)\,\mathrm{P}(\mathbf{w_2}|\mathcal{L}_2)$ for all $\mathbf{w_1} \in \mathcal{H}_1$ and $\mathbf{w_2} \in \mathcal{H}_2$, then

$$\mathrm{P}(r|w, \mathcal{L}) = \sum_{k=0}^{r} \mathrm{P}(r-k|w, \mathcal{L}_1)\,\mathrm{P}(k|w, \mathcal{L}_2). \tag{21}$$

More generally, if $\mathcal{L}$ can be divided into M mutually independent parts, then equation 21 can be iteratively applied $M-1$ times to determine $\mathrm{P}(r|w, \mathcal{L})$. In the speech recognition example, $\mathcal{H}$ might correspond to a long passage while each $\mathcal{H}_i$ could be hypotheses for the ith sentence in the passage where we assume sentences are independent of each other.

This points the way to an efficient algorithm for calculating $\widetilde{\mathrm{P}}$ directly from word lattices, which we defer to future research.

3.2.2 Unsupervised MAP adaptation using lattices

In contrast to the unsupervised adaptation approach used in section 3.1, where the in-domain counts ($c^I_{s,i}$ and C^I_s in equation 3) were obtained from the transcripts, here the Monte Carlo approach of equation 19 is used and the in-domain counts are found as

$$c^I_{s,i} = \frac{1}{M} \sum_{n=1}^{M} |\mathbf{w_n}| \mathrm{P}^{\mathbf{w_n}}(s, w_i) \tag{22}$$

and, as before,

$$C^I_s = \sum_i c^I_{s,i}. \tag{23}$$

Motivated by the experimental results in section 3.1, we initially decided to use a count merging approach for adaptation. In other words, we intended to use the prior parameterization as in equation 5. However, we found an important distinction between supervised or small sample unsupervised MAP adaptation with unsupervised MAP adaptation on a large sample. The intuition behind MAP adaptation in the supervised case is: with few in-domain observations, the model should be close to the out-of-domain trained model; as more in-domain observations are obtained, the model should move toward the maximum likelihood model given the observations. In other words, as the in-domain counts grow, they should swamp the out-of-domain counts. However, for unsupervised adaptation, the out-of-domain model, being based upon supervised annotations, constrains the noisy in-domain observations, and hence

provides a benefit even when the amount of unlabeled data is very large. For this reason, we modified the parameterization of these MAP experiments to take this effect into account. Instead of using the prior distribution parameters as in equation 4, we here use

$$\nu_{s,i} = \frac{\alpha}{\beta} \frac{\sum_{n=1}^{M} |\mathbf{w}_n|}{M|\mathbf{w}_0|} C_s^O \mathrm{P}^O(\omega_i \mid s) + 1 \tag{24}$$

so that larger unlabeled sample sizes result in greater compensatory scaling of the out-of-domain observations, to avoid them being swamped. For the empirical results reported in the next section, $\frac{\alpha}{\beta} = 3.5$. This resulted in good performance for both word-lattice sampling and one-best adaptation approaches.

3.2.3 Experimental results

We evaluated both Monte Carlo word lattice sampling and one-best ASR transcription adaptation scenarios by measuring word accuracy within an AT&T "How May I Help You?" spoken dialog application known as Customer Care (CC) [13]. The out-of-domain corpus was 171,343 words transcribed from a previously deployed application known as Operator Services (OS) [13]. The range of topics served by the CC application is disjoint from those served by the OS application. The baseline language model is a trigram built from the above corpus, with 3337 unigram, 40821 bigram, and 23360 trigram probabilities after shrinking.

In the CC domain, we have a 2000 utterance manually transcribed test set, and approximately 1050 hours of untranscribed in-domain utterances for unsupervised adaptation. For all of the results, we produce word lattices or one-best transcriptions for the untranscribed in-domain utterances. Using the counts obtained from those transcriptions, the OS baseline model is adapted using the MAP algorithm with the prior distribution as described in the previous section.

The experiments compare word accuracy results for two trials: word lattice sampling with 1000 samples and one-best transcription. We varied the amount of untranscribed data provided to the training algorithm, to see its effect. Figure 1 plots the results versus the baseline from zero to 300 hours of in-domain unlabeled data. In figure 2 the results for all of the 1050 hours of unlabeled data are shown, showing the data size on a log scale. With all of the unlabeled data included, both methods provide a 3.7 percent improvement in word accuracy over the baseline model. The Good-Turing sampling approach converges more quickly than the one-best approach, providing a 0.6 percent advantage when the amount of unlabeled data is limited to 57 hours. In the limit, however, the one-best reaches the same level of performance.

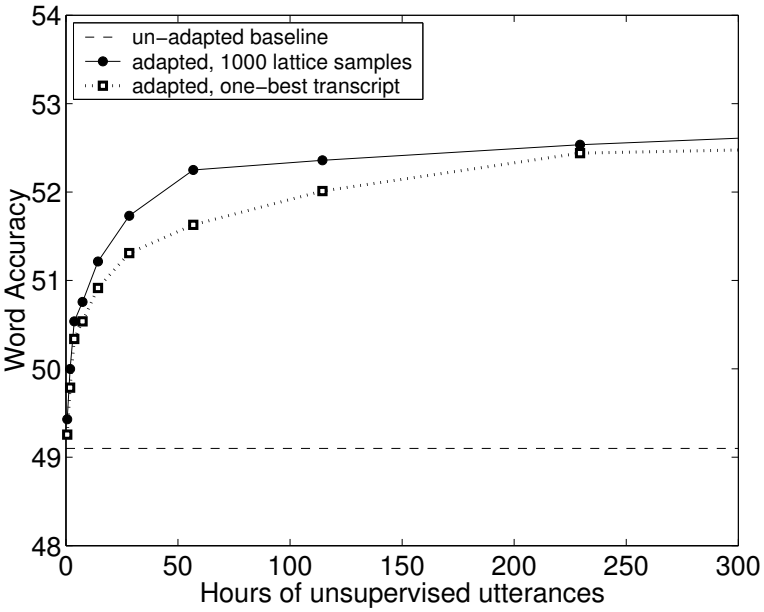

Fig. 1. Word Accuracy versus hours of unlabeled utterances for adaptation based on word lattice sampling and ASR one-best output, for up to 300 hours of in-domain utterances.

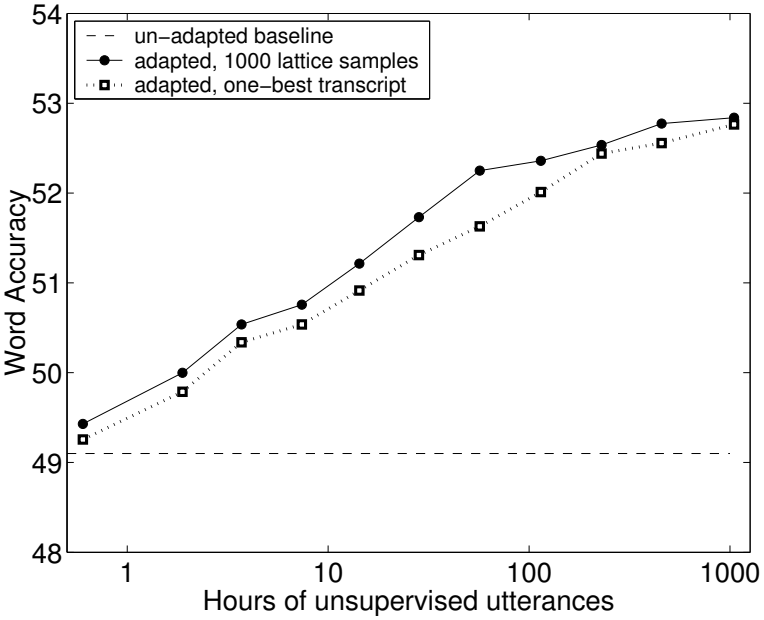

Fig. 2. Word Accuracy versus hours of unlabeled utterances for adaptation based on word lattice sampling and ASR one-best output, with the hours plotted in log scale.

For the purpose of perplexity computation, we mapped out-of-vocabulary words (3.5 percent of tokens in the test set) to an unknown token, which was reserved a unigram probability of 0.00001. Figure 3 plots perplexity versus hours of unlabeled training data, with the hours in log scale. From this we can see that, indeed, the one-best approach catches up with the lattice sam-

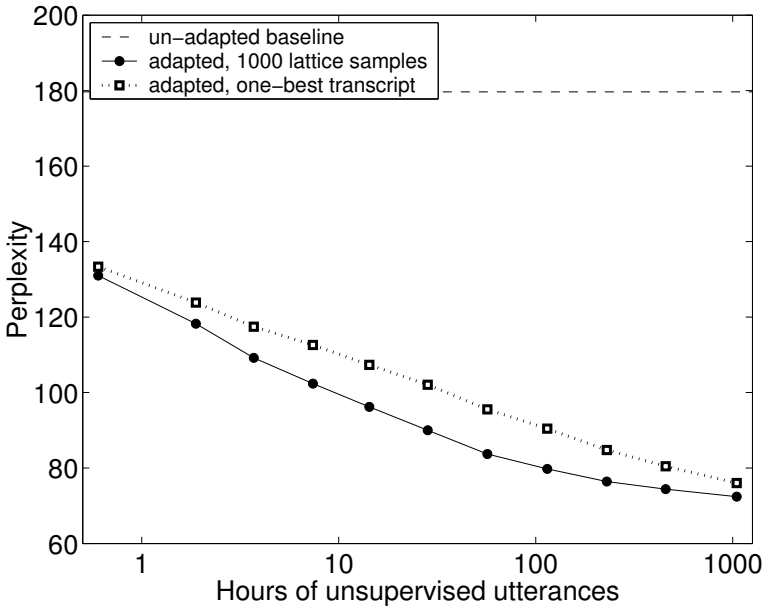

Fig. 3. Perplexity versus hours of unlabeled utterances for adaptation based on word lattice sampling and ASR one-best output, with the hours plotted in log scale.

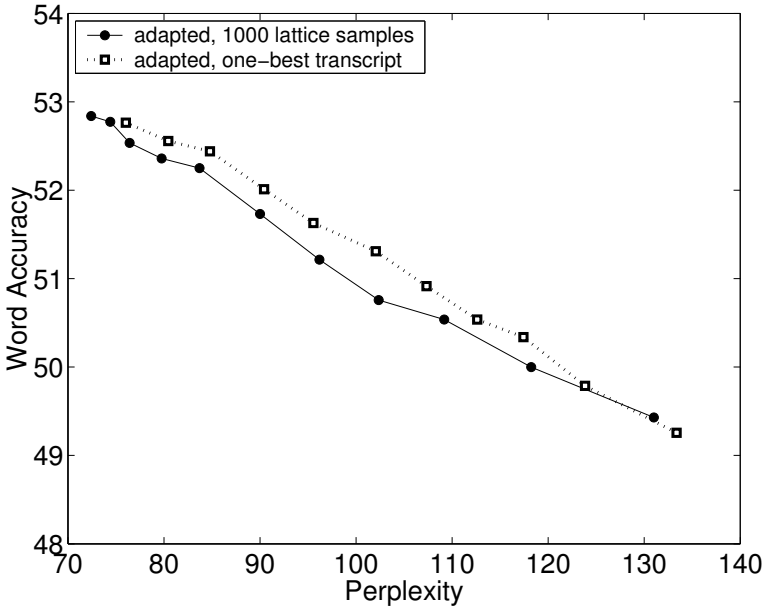

Fig. 4. Word Accuracy versus Perplexity for adaptation based on word lattice sampling and ASR one-best output.

pling approach in terms of perplexity as well as word accuracy. Figure 4 plots perplexity versus word accuracy. The one-best approach provides somewhat better word accuracy than the lattice sampling approach at the same perplexity level, indicating that some of the improvements in modeling provided by the lattice sampling are not particularly useful to the recognizer.

To verify that these results hold up in different domains, we revisited the SS-

NIFR voicemail adaptation from section 3.1. In this domain, we found results consistent with the results presented in this section. Unsupervised adaptation over the 17 hour training set by sampling word lattices yielded a 0.2 percent improvement over adaptation using the ASR transcripts.

3.2.4 Conclusions

The one-best results in the last section are consistent with the results in section 3.1 in that unsupervised MAP adaptation provides accuracy gains even though the baseline accuracy for this task is much lower. The results on this task show that in addition, there is a small, consistent gain to be had by using the word lattices, when the amount of unlabeled data is limited. These results also show that when presented with a very large adaptation sample, there is no advantage of using lattices instead of transcripts.

4 MAP Adaptation of PCFGs

The MAP formulation presented in section 2 directly applies to many common models used in natural language processing, including n-tag part-of-speech (POS) tagging models, finite-state shallow parsing models, or probabilistic context-free grammars (PCFGs). In these models, the states s in equation 3 correspond to conditioning variables and the weights $\omega_{s,i}$ are the conditional probability estimates for the conditioned variables. For example, in a bi-tag POS tagger, the probability of each POS tag t_k is conditioned on the previous tag t_{k-1}, i.e. $s = t_{k-1}$, and $\omega_{s,i} = \mathrm{P}(t_k \mid t_{k-1})$.

In this section the MAP adaptation framework is applied to a PCFG used by a statistical parser. The PCFG, parser and the counts involved in the MAP adaptation process are first described in section 4.1, then experimental results are described in section 4.2.

4.1 Grammar and parser

A context-free grammar (CFG) $G = (V, T, P, S^\dagger)$, consists of a set of non-terminal symbols V, a set of terminal symbols T, a start symbol $S^\dagger \in V$, and a set of rule productions P of the form: $A \rightarrow \gamma$, where $A \in V$ and $\gamma \in (V \cup T)^*$. A probabilistic context-free grammar (PCFG) is a CFG with a probability assigned to each rule, such that the probabilities of all rules expanding a given non-terminal sum to one; specifically, each right-hand side

has a probability given the left-hand side of the rule[3].

The MAP formulation in equation 3 directly applies to PCFGs. Let s denote the left-hand side of a production, and $\omega_{s,i}$ the i-th possible expansion of s. The MAP estimate for the probability of that expansion, $\mathrm{P}(\omega_{s,i} \mid s)$, is obtained from equation 3. The $c^I_{s,i}$ and $c^O_{s,i}$ counts involved in that estimation are obtained from the frequencies of $\omega_{s,i}$ expansions following a left-hand side s in the in-domain and out-of-domain samples respectively. Similarly, the C^I_s and C^O_s counts are obtained from the frequencies of observing the left-hand side s in either domain. Here, as in MAP adaptation of n-gram models, various choices of the parameterization of the prior distribution exist.

For the empirical trials, we used a top-down, left-to-right (incremental) statistical beam-search parser [29,31]. We refer readers to the cited papers for details on this parsing algorithm. Briefly, the parser maintains a set of candidate analyses, each of which is extended to attempt to incorporate the next word into a fully connected partial parse. As soon as "enough" candidate parses have been extended to the next word, all parses that have not yet attached the word are discarded, and the parser moves on to the next word. This beam search is parameterized with a base beam parameter δ, which controls how many or how few parses constitute "enough". Candidate parses are ranked by a figure-of-merit, which promotes better candidates, so that they are worked on earlier. The figure-of-merit consists of the probability of the parse to that point times a look-ahead statistic, which is an estimate of how much probability mass it will take to connect the parse with the next word. It is a generative parser that does not require any pre-processing, such as POS tagging or chunking. It has been demonstrated in the above papers to perform competitively on standard statistical parsing tasks with full coverage. Baseline results below will provide a comparison with other well known statistical parsers.

The PCFG is a *Markov* grammar [6,4], i.e. the production probabilities are estimated by decomposing the joint probability of the categories on the right-hand side into a product of conditionals via the chain rule, and making a Markov assumption. Thus, for example, a first order Markov grammar conditions the probability of the category of the i-th child of the left-hand side on the category of the left-hand side and the category of the $(i-1)$-th child of the left-hand side. The benefits of Markov grammars for a top-down parser of the sort we are using is detailed in [31]. Further, as in [29,31], the production probabilities are conditioned on the label of the left-hand side of the production, as well as on features from the left-context. The model is smoothed using

[3] An additional condition for well-formedness is that the PCFG is consistent or tight, i.e. there is no probability mass lost to infinitely large trees. [5] proved that this condition is met if the rule probabilities are estimated using relative frequency estimation from a corpus.

Features for non-POS left-hand sides	Features for POS left-hand sides
0 Left-hand side (LHS)	0 Left-hand side (LHS)
1 Last child of LHS	1 Parent of LHS (PAR)
2 2nd last child of LHS	2 Last child of PAR
3 3rd last child of LHS	3 Parent of PAR (GPAR)
4 Parent of LHS (PAR)	4 POS of C-Commanding head
5 Last child of PAR	5 C-Commanding lexical head
6 Parent of PAR (GPAR)	6 Next C-Commanding lexical head
7 Last child of GPAR	
8 First child of conjoined category	
9 Lexical head of current constituent	

Table 8
Conditioning features for the probabilistic CFG used in the reported empirical trials

standard deleted interpolation, wherein a mixing parameter μ is estimated using EM on a held out corpus, such that probability of a production $A \rightarrow \gamma$, conditioned on j features from the left context, $X_1^j = X_1 \ldots X_j$, is defined recursively as

$$\begin{aligned} \mathrm{P}(A \rightarrow \gamma \mid X_1^j) &= \mathrm{P}(\gamma \mid A, X_1^j) \\ &= (1-\mu)\widehat{\mathrm{P}}(\gamma \mid A, X_1^j) + \mu \mathrm{P}(\gamma \mid A, X_1^{j-1}) \end{aligned} \tag{25}$$

where $\widehat{\mathrm{P}}$ is the maximum likelihood estimate of the conditional probability. For MAP adaptation purposes, $s = A, X_i^j$. These conditional probabilities decompose via the chain rule as mentioned above, and a Markov assumption limits the number of previous children already emitted from the left-hand side that are conditioned upon. These previous children are treated exactly as other conditioning features from the left context. Table 8 gives the conditioning features that were used for all empirical trials in this paper. There are different conditioning features for parts-of-speech (POS) and non-POS non-terminals. Deleted interpolation leaves out one feature at a time, in the reverse order as they are presented in the table 8. See [31] for more details on the parsing approach.

The PCFG used for these trials was induced using relative frequency estimation from a transformed treebank. The trees are transformed with a selective left-corner transformation [19] that has been flattened as presented in [30]. This transform is only applied to left-recursive productions, i.e. productions of the form $A \rightarrow A\gamma$. The transformed trees look as in figure 5. The transform has the benefit for a top-down incremental parser of this sort of delaying many

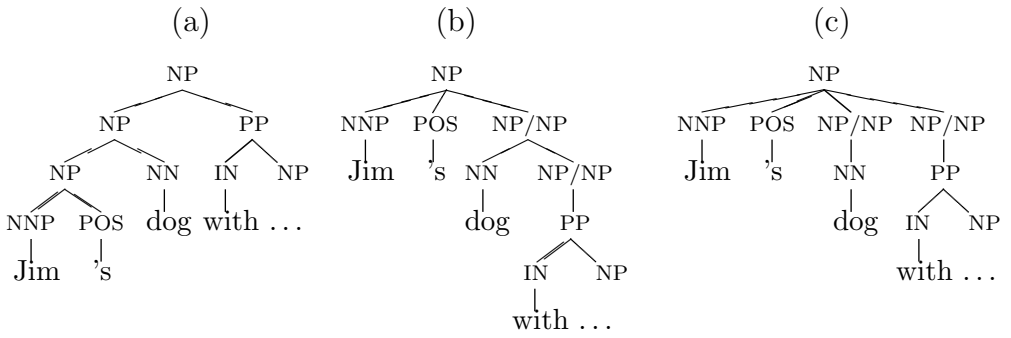

Fig. 5. Three representations of NP modifications: (a) the original treebank representation; (b) Selective left-corner representation; and (c) a flat structure that is unambiguously equivalent to (b)

of the parsing decisions until later in the string, without unduly disrupting the immediate dominance relationships that provide conditioning features for the probabilistic model. The parse trees that are returned by the parser are then de-transformed to the original form of the grammar for evaluation [4].

For the trials reported in the next section, the base beam parameter is set at $\delta = 10$. In order to avoid being pruned, a parse must be within a probability range of the best scoring parse that has incorporated the next word. Let k be the number of parses that have incorporated the next word, and let $\widetilde{\mathrm{P}}$ be the best probability from among that set. Then the probability of a parse must be above $\frac{\widetilde{\mathrm{P}}_{k^3}}{10^{\delta}}$ to avoid being pruned.

4.2 Experimental Results

To evaluate MAP adapted PCFGs for statistical parsing, we experimented with two corpora from the Penn Treebank II: the Wall St. Journal treebank (WSJ) and the treebank of the Brown corpus. Adaptation in both directions was evaluated: from WSJ to Brown and vice versa. For the Wall St. Journal portion, we used the standard breakdown: sections 2-21 were kept for training data; section 24 was held-out development data; and section 23 was for evaluation. For the Brown corpus portion, we obtained the training and evaluation sections used in [11]. In that paper, no held-out section was used for parameter tuning [5], so we further partitioned the training data into kept and held-out data. The sizes of the corpora are given in table 9, as well as labels that are used to refer to the corpora in subsequent tables.

[4] See [18] for a presentation of the transform/de-transform paradigm in parsing.
[5] According to the author, smoothing parameters for his parser were based on the formula from [7].

	Wall St. Journal			Brown Corpus		
	Training	Held out	Eval	Training	Held out	Eval
Label	WSJ;2-21	WSJ;24	WSJ;23	Brown;T	Brown;H	Brown;E
Sentences	39,832	1,346	2,416	19,740	2,078	2,425
Words	950,028	32,853	56,684	373,152	40,046	45,950

Table 9
Corpus sizes

System	Training	Heldout	LR	LP	F
Gildea	WSJ;2-21		80.3	81.0	80.6
MAP	WSJ;2-21	WSJ;24	81.3	80.9	81.1
MAP	WSJ;2-21	Brown;H	81.6	82.3	81.9
Gildea	Brown;T,H		83.6	84.6	84.1
MAP	Brown;T	Brown;H	84.4	85.0	84.7

Table 10
Parser performance on Brown;E, baselines. Note that the Gildea results are for sentences $\leq$ 40 words in length.

4.2.1 Baseline performance

The first results are for parsing the Brown corpus. Table 10 presents our baseline performance, compared with the Gildea (2001) results. Our system is labeled as 'MAP'. All parsing results are presented as labeled precision (LP) and labeled recall (LR), as well as F-measure, which is defined as $2(LR)(LP)/(LR+LP)$. Whereas [11] reported parsing results just for sentences of length less than or equal to 40, our results are for all sentences. The goal is not to improve upon Gildea's parsing performance, but rather to try to get more benefit from the out-of-domain data. While our performance is 0.5-1.5 percent better than Gildea's, the same trends hold – low eighties in accuracy when using the Wall St. Journal (out-of-domain) training; mid eighties when using the Brown corpus training. Notice that using the Brown held out data with the Wall St. Journal training improved precision substantially. Tuning the parameters on in-domain data can make a big difference in parser performance. Choosing the smoothing parameters as Gildea did, based on the distribution within the corpus itself, may be effective when parsing within the same distribution, but appears less so when using the treebank for parsing outside of the domain.

Table 11 gives the baseline performance on section 23 of the WSJ Treebank. Note, again, that the Gildea results are for sentences $\leq$ 40 words in length, while all others are for all sentences in the test set. Also, Gildea did not report performance of a Brown corpus trained parser on the WSJ. Our performance under that condition is not particularly good, but again using an in-domain

System	Training	Heldout	LR	LP	F
MAP	Brown;T	Brown;H	76.0	75.4	75.7
MAP	Brown;T	WSJ;24	76.9	77.1	77.0
Gildea	WSJ;2-21		86.1	86.6	86.3
MAP	WSJ;2-21	WSJ;24	86.9	87.1	87.0
[3]	WSJ;2-21	WSJ;24	86.7	86.6	86.6
[27]	WSJ;2-21		86.3	87.5	86.9
[7]	WSJ;2-21		88.1	88.3	88.2
[4]	WSJ;2-21	WSJ;24	89.6	89.5	89.5
[8]	WSJ;2-21		89.6	89.9	89.7

Table 11
Parser performance on WSJ;23, baselines. Note that the Gildea results are for sentences ≤ 40 words in length. All others include all sentences.

Training set size (%)	100	75	50	25	10	5
LR	86.9	86.6	86.3	84.8	82.6	80.4
LP	87.1	86.8	86.4	85.0	82.6	80.6
F-measure	87.0	86.7	86.4	84.9	82.6	80.5

Table 12
Performance of the MAP parser on WSJ;23, trained on variously sized subsets of WSJ;2-21 and using WSJ;24 as the held out set

held out set for parameter tuning provided a substantial increase in accuracy, somewhat more in terms of precision than recall. Our baseline results for a WSJ section 2-21 trained parser are slightly better than the Gildea parser, at more-or-less the same level of performance as [3] and [27], but several points below the best reported results on this task.

Table 12 gives the MAP baseline performance on WSJ;23, with models trained on fractions of the entire 2-21 test set. Sections 2-21 contain approximately 40,000 sentences, and we partitioned them by percentage of total sentences. From table 12 we can see that parser performance degrades quite dramatically when there is less than 20,000 sentences in the training set, but that even with just 2000 sentences, the system outperforms one trained on the Brown corpus (80.5% vs. 77.0% F-measure).

System	Training	Adapt	Prior	Param	Performance F (LR, LP)	ΔF
Gildea	WSJ;2-21	Brown;T,H	Merge	$\frac{\alpha}{\beta} = 1$	84.35 (83.9,84.8)	0.25
MAP	WSJ;2-21	Brown;T	Merge	$\frac{\alpha}{\beta} = 1$	85.25 (84.9,85.6)	0.55
MAP	WSJ;2-21	Brown;T	Merge	$\frac{\alpha}{\beta} = 0.2$	85.65 (85.4,85.9)	0.95
MAP	WSJ;2-21	Brown;T	Interp.	$\lambda = 0.25$	85.60 (85.3,85.9)	0.90

Table 13
Parser performance on Brown;E, supervised adaptation. The baseline performance for the Gildea parser is $84.1(83.6, 84.6)$ and that parser did not use a held out set. The baseline performance for the MAP parser is $84.7(84.4, 85.0)$ and used Brown;H as the held out set.

4.2.2 Supervised adaptation

Analogous to the n-gram experiments reported in section 3.1.2 we first investigate the effect of using different parameterizations of the prior distribution in a supervised MAP adaptation approach. Both a count merging and model interpolation approach were evaluated. Table 13 presents parsing results on the Brown;E test set for models using both in-domain and out-of-domain training data. The table gives the adaptation (in-domain) treebank that was used, and the parameterization of the prior distribution. [11] merged the two corpora, which simply adds the counts from the out-of-domain treebank to the in-domain treebank, i.e. uses equation 5 with $\frac{\alpha}{\beta} = 1$. This resulted in a 0.25 improvement in the F-measure. In our case, combining the counts in this way yielded 0.5%, perhaps because of the in-domain tuning of the smoothing parameters. However, when we optimize $\frac{\alpha}{\beta}$ empirically on the held-out corpus, we can get nearly a 1% improvement. Model interpolation in this case performs nearly identically to count merging.

Adaptation to the Brown corpus, however, does not adequately represent what is likely to be the most common adaptation scenario, i.e. adaptation to a consistent domain with limited in-domain training data. The Brown corpus is not really a domain; it was built as a balanced corpus, and hence is the aggregation of multiple domains. The reverse scenario – Brown corpus as out-of-domain parsing model and Wall St. Journal as novel domain – is perhaps a more natural one. In this direction, [11] also reported very small improvements when adding in the out-of-domain treebank. This may be because of the same issue as with the Brown corpus, namely that the optimal ratio of in-domain to out-of-domain is not 1 and the smoothing parameters need to be tuned to the new domain; or it may be because the new domain has a million words of training data, and hence has less use for out-of-domain data. To tease these apart, we partitioned the WSJ training data (sections 2-21) into smaller treebanks, and looked at the gain provided by adaptation as the in-domain

System	Fraction of WSJ;2-21 (%)	$\frac{\alpha}{\beta}$	Trained F (LR,LP)	Adapted F (LR,LP)	ΔF
Gildea	100	1	86.35 (68.1,86.6)	86.60 (86.3,86.9)	0.25
MAP	100	0.2	87.00 (86.9,87.1)	87.35 (87.2,87.5)	0.35
MAP	75	0.2	86.7 (86.6,86.8)	87.2 (87.1,87.3)	0.5
MAP	50	0.2	86.35 (86.3,86.4)	86.8 (86.7,86.9)	0.45
MAP	25	0.2	84.9 (84.8,85)	85.4 (85.3,85.5)	0.5
MAP	10	0.2	82.6 (82.6,82.6)	84.35 (84.3,84.4)	1.75
MAP	10	1	82.6 (82.6,82.6)	83.3 (83.2,83.4)	0.7
MAP	5	0.2	80.5 (80.4,80.6)	83.05 (83,83.1)	2.55

Table 14
Parser performance on WSJ;23, supervised adaptation. All models use Brown;T,H as the out-of-domain treebank. Trained models are built from the fractions of WSJ;2-21, with no out-of-domain treebank. The performance of the parser trained on the Brown;T corpus using WSJ;24 as heldout data is 77.0(76.9, 77.1).

observations grow. These smaller treebanks provide a more realistic scenario: rapid adaptation to a novel domain will likely occur with far less manual annotation of trees within the new domain than can be had in the full Penn Treebank.

Table 14 presents parsing accuracy when a model trained on the Brown corpus is adapted with part or all of the WSJ training corpus. It compares the performance of an adapted parser with one obtained by training on the same sample (i.e. table 12). Performance improvements over the parser trained on Brown;T, using the WSJ;24 section as the heldout set range from 6.05% F-measure using 5% of the adaptation set to 10.35% using the entire set.

From this point forward, we only present results for count merging, since model interpolation consistently performed 0.2-0.5% below the count merging approach, which is consistent with the results on n-gram adaptation. The $\frac{\alpha}{\beta}$ mixing ratio was empirically optimized on the held out set when the in-domain training was just 10% of the total; this optimization makes over 1% difference in accuracy. Like Gildea, with large amounts of in-domain data, adaptation improved our performance by 0.5% or less. When the amount of in-domain data is small, however, the impact of adaptation is much greater.

For the unsupervised adaptation experiments of n-gram models, the in-domain data was automatically annotated using the output of a speech recognizer using the out-of-domain n-gram model. Here, we use the parsing model trained on out-of-domain data, and output a set of candidate parse trees for the strings in the in-domain corpus, with their normalized scores. These normalized scores (posterior probabilities) are then used to give weights to the features extracted from each candidate parse, in just the way that they provide expected counts for an expectation maximization algorithm.

For the unsupervised trials that we report, we collected up to 20 candidate parses per string [6] . We were interested in investigating the effects of adaptation, not in optimizing performance, hence we did not empirically optimize the mixing parameter $\frac{\alpha}{\beta}$ for the new trials, so as to avoid obscuring the effects due to adaptation alone. Rather, we used the best performing parameter from the supervised trials, namely 0.2. Since we are no longer limited to manually annotated data, the amount of in-domain WSJ data that we can include is essentially unlimited. Hence the trials reported go beyond the 40,000 sentences in the Penn WSJ Treebank, to include up to 5 times that number of sentences from other years of the WSJ.

Table 15 shows the results of unsupervised adaptation as we have described it. Note that these improvements are had without seeing any manually annotated Wall St. Journal treebank data. Using the approximately 40,000 sentences in f2-21, we derived a 3.8 percent F-measure improvement over using just the out-of-domain data. This is 37% of the 10.35% gain obtained by supervised adaptation on that same sample. Going beyond the size of the Penn Treebank, we continued to gain in accuracy, reaching a total F-measure improvement of 4.2 percent with 200 thousand sentences, approximately 5 million words. A second iteration with this best model, i.e. re-parsing the 200 thousand sentences with the adapted model and re-training, yielded an additional 0.65 percent F-measure improvement, for a total F-measure improvement of 4.85 percent over the baseline model.

A final unsupervised adaptation scenario that we investigated is self-adaptation, i.e. adaptation on the test set itself. Because this adaptation is completely unsupervised, thus does not involve looking at the manual annotations at all, it can be equally well applied using the test set as the unsupervised adaptation set. Using the same adaptation procedure presented above on the test set itself, i.e. producing the top 20 candidates from WSJ;23 with normalized posterior probabilities and re-estimating, we produced a self-adapted parsing

[6] Because of the left-to-right, heuristic beam-search, the parser does not produce a chart, rather a set of completed parses.

Adaptation Sentences	Iteration	Performance F (LR,LP)	ΔF
0	0	75.70 (76.0,75.4)	
4000	1	78.25 (78.6,77.9)	2.55
10000	1	78.45 (78.9,78.0)	2.75
20000	1	78.90 (79.3,78.5)	3.20
30000	1	79.30 (79.7,78.9)	3.60
39832	1	79.50 (79.9,79.1)	3.80
100000	1	79.45 (79.7,79.2)	3.75
200000	1	79.90 (80.2,79.6)	4.20
200000	2	80.55 (80.6,80.5)	4.85

Table 15
Parser performance on WSJ;23, unsupervised adaptation. For all trials, the base training is Brown;T, the held out is Brown;H plus the parser output for WSJ;24, and the mixing parameter $\frac{\alpha}{\beta}$ is 0.20.

model. This yielded an F-measure accuracy of 76.8, which is a 1.1 percent improvement over the baseline.

4.4 Conclusions

The MAP domain adaptation results for PCFG grammars is consistent with the results obtained for n-gram models. Use of either supervised or unsupervised adaptation improves accuracy over an unadapted system. This shows that the MAP framework is applicable to models other than n-grams alone and that the adaptation results are not strongly dependent on the type of model that is adapted.

5 Discussion

This paper has shown that the MAP framework directly applies to the domain adaptation problem for both n-gram and PCFG models. Experimental results show that MAP adaptation to a new domain provides accuracy improvements using either supervised adaptation, unsupervised adaptation or a hybrid approach.

The n-gram adaptation results in section 3.1 and PCFG adaptation results

in section 4.2 seem to suggest that the prior distribution parameterization corresponding to a count merging approach slightly outperforms that of a model interpolation approach. The n-gram results also show that on a 17 hour sample, about half of the supervised accuracy improvement was obtained by unsupervised adaptation. For PCFG adaptation this fraction was about one third.

The results on a much larger adaptation sample reported in 3.2 show that for moderate adaptation samples (about 50 hours) there is an advantage in using lattice-based adaptation as opposed to transcription based adaptation. It also shows that this advantage disappears with an increasing adaptation sample, showing no benefits at a 1050 hour adaptation sample.

Adaptation using transcripts is simple, so this is likely to be the method of choice in the case when the amount of unlabeled data is large. Since there is no manual labeling required, in many cases it should be possible to generate essentially arbitrary amounts of training data for a given domain. However, when the amount of unlabeled data is limited, a lattice-based approach seems more attractive, as it provides faster accuracy improvements with increasing adaptation sample size.

The general MAP formulation allowed us, in the unsupervised case, to use a more effective parameterization of the adaptation. It may well be that, in certain circumstances, more fine-grained control of the adaptation parameters, which take into account the specific conditioning state s, will yield improvements over the simple count merging or model interpolation parameterizations. This formulation provides the framework within which such explorations can take place.

References

[1] M. Bacchiani. Automatic transcription of voicemail at AT&T. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001.

[2] M. Bacchiani and B. Roark. Unsupervised language model adaptation. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 224–227, 2003.

[3] E. Charniak. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 598–603, 1997.

[4] E. Charniak. A maximum-entropy-inspired parser. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics*, pages 132–139, 2000.

[5] Z. Chi and S. Geman. Estimation of probabilistic context-free grammars. *Computational Linguistics*, 24(2):299–305, 1998.

[6] M. J. Collins. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23, 1997.

[7] M. J. Collins. *Head-Driven Statistical Models for Natural Language Parsing.* PhD thesis, University of Pennsylvania, 1999.

[8] M. J. Collins. Discriminative reranking for natural language parsing. In *The Proceedings of the 17th International Conference on Machine Learning*, 2000.

[9] M. J. F. Gales. Maximum likelihood linear transformations for hmm-based speech recognition. *Computer Speech and Language*, pages 75–98, 1998.

[10] J.-L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Transactions on Speech and Audio Processing*, 2(2):291–298, 1994.

[11] D. Gildea. Corpus variation and parser performance. In *Proceedings of the Sixth Conference on Empirical Methods in Natural Language Processing (EMNLP-01)*, 2001.

[12] I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrica V*, 40(3,4):237–264, 1953.

[13] A. L. Gorin, A. Abella, T. Alonso, G. Riccardi, and J. H. Wright. Automated natural spoken dialogue. *IEEE Computer Magazine*, 35(4):51–56, 2002.

[14] R. Gretter and G. Riccardi. On-line learning of language models with word error probability distributions. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 557–560, 2001.

[15] R. Hwa. Supervised grammar induction using training data with limited constituent information. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 1999.

[16] R. Hwa. On minimizing training corpus for parser acquisition. In *Proceedings of the Fifth Computational Natural Language Learning Workshop*, 2001.

[17] F. Jelinek and R. L. Mercer. Interpolated estimation of markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397, 1980.

[18] M. Johnson. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):617–636, 1998.

[19] M. Johnson and B. Roark. Compact non-left-recursive grammars using the selective left-corner transform and factoring. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING)*, pages 355–361, 2000.

[20] S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recogniser. *IEEE Transactions on Acoustic, Speech, and Signal Processing*, 35(3):400–401, 1987.

[21] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 181–184, 1995.

[22] L. Lamel, J.-L. Gauvain, and G. Adda. Unsupervised acoustic model training. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 877–880, 2002.

[23] C. J. Legetter and P. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech and Language*, pages 171–185, 1995.

[24] A. Ljolje, D. Hindle, M. Riley, and R. Sproat. The AT&T LVCSR-2000 system. In *Proceedings of the NIST LVCSR Workshop*, 2000.

[25] L. Mangu, E. Brill, and A. Stolcke. Finding consensus among words: Lattice-based word error minimization. In *Proceedings of Eurospeech*, 1999.

[26] F. C. Pereira and Y. Schabes. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 128–135, 1992.

[27] A. Ratnaparkhi. Learning to parse natural language with maximum entropy models. *Machine Learning*, 34:151–175, 1999.

[28] M. Riley, B. Roark, and R. Sproat. Good-turing estimation from word lattices for unsupervised language model adaptation. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 453–458, 2003.

[29] B. Roark. Probabilistic top-down parsing and language modeling. *Computational Linguistics*, 27(2):249–276, 2001.

[30] B. Roark. *Robust Probabilistic Predictive Syntactic Processing*. PhD thesis, Brown University, 2001. http://arXiv.org/abs/cs/0105019.

[31] B. Roark. Robust garden path parsing. *Natural Language Engineering*, 10(1):1–24, 2004.

[32] B. Roark and M. Bacchiani. Supervised and unsupervised PCFG adaptation to novel domains. In *Proceedings of the Human Language Technology Conference and Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 205–212, 2003.

[33] A. Stolcke. Error modeling and unsupervised language modeling. In *Proceedings of the 2001 NIST Large Vocabulary Conversational Speech Recognition Workshop*, Linthicum, Maryland, May 2001.

[34] I. H. Witten and T. C. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4):1085–1094, 1991.

[35] P. Woodland and T. Hain. The September 1998 HTK Hub 5E System. In *The Proceedings of the* 9^{th} *Hub-5 Conversational Speech Recognition Workshop*, 1998.

[36] P. Woodland, T. Hain, G. Moore, T. Niesler, D. Povey, A. Tuerk, and E. Whittaker. The 1998 HTK broadcast news transcription system: Development and results. In *DARPA Broadcast News Transcription and Understanding Workshop*, 1998.