# GOOD-TURING ESTIMATION FROM WORD LATTICES FOR UNSUPERVISED LANGUAGE MODEL ADAPTATION

*Michael Riley*[†]*, Brian Roark*[†]*, and Richard Sproat*[‡]

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

In this paper we present a comparison of using the weighted word lattice output of a recognizer versus its one-best transcription for unsupervised language model adaptation. We begin with a general analysis of how to smooth word probabilities when the sample is *hidden* as is the case with recognizer lattices. For each smoothing technique for the known sample case, we show there is a natural generalization to the hidden case. In particular, we use this generalization with the well-known Good-Turing estimate on word lattices, and show results using Monte Carlo methods for building Katz backoff models. In our realistic adaptation task, with mismatched acoustic and language models, we find that Katz backoff models trained on word lattice samples provide a small, consistent benefit over those trained on one-best output, most notably when there is a limited amount of adaptation data (less than 100 hours). Thus, while the recognizer one-best transcription can provide an effective approximation for the purpose of language model adaptation under certain circumstances, the word lattice provides information that can be exploited for more robust language modeling.

## 1. INTRODUCTION

As automatic speech recognition (ASR) becomes more widely used, often as a key piece of larger applications, language model (LM) and acoustic model (AM) adaptation to novel domains is indispensable. A common adaptation scenario involves porting to domains with little or no manually annotated data, but perhaps with a relatively large amount of unannotated speech data. Research into unsupervised LM adaptation [1, 2, 3] has relied upon ASR transcripts, rather than weighted word lattices, which provide a probability distribution over a number of competing hypothesis transcriptions, and hence contain more information that could potentially be exploited for adaptation. 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 techniques, such as deleted interpolation [4] and Witten-Bell smoothing [5] could straight-forwardly use these counts, since these techniques rely on mixing maximum likelihood (relative frequency) estimates. However, other techniques, such as Katz backoff [6] or Kneser-Ney estimation [7], which are more widely used, rely on integral counts in their definition and exactly how to generalize them is not immediate. For this reason, we take the time here to outline a general approach to extending arbitrary smoothing techniques to lattice corpora. Using Good-Turing smoothing, we then employ a Monte Carlo version of our generalization for Katz backoff modeling and compare to the now standard technique of using one-best transcription.

After presenting our generalization for use with word lattices and our method for unsupervised n-gram model adaptation, we provide empirical results for adaptation to a novel customer call classification application. 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. 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.

The contribution of this paper is twofold. First, it provides a generalization of arbitrary LM smoothing techniques

to the word lattice case and, in particular, applies this to Katz estimation. Second, it documents a comparison between using word lattices and the more common technique of using one-best ASR transcription for unsupervised language model adaptation, demonstrating that, under certain common circumstances, there can be a small gain by using the former.

## 2. 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.

### 2.1. Known random sample

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

$$\hat{p}_{ml}(w|\mathbf{w}) = \frac{r}{N} \tag{1}$$

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 [8].

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

$$\hat{p}_g(w|\mathbf{w}) = \frac{r+1}{N+1} \frac{\mathcal{E}_{N+1}[n_{r+1}]}{\mathcal{E}_N[n_r]} \tag{2}$$

where $n_r$ is the number of distinct words that have count $r$ in a sample [8]. Good shows that his estimate for the probability of a word $w$ is equal to $\mathcal{E}[p_u | c(u, \mathbf{w}) = c(w, \mathbf{w})]$, 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:

$$\hat{p}_t(w|\mathbf{w}) = \frac{r+1}{N} \frac{n_{r+1}}{n_r} \tag{3}$$

approximates $\mathcal{E}_N[n_r]$ with $n_r$ from the sample [8].

Katz shows how to apply the Good-Turing estimate not just to words but to n-grams of words for building a stochastic LM [6].

### 2.2. Hidden random sample

Consider now that the sample is also unknown to us. Instead we are only given $\mathcal{L} = (\mathcal{H}, p(\mathbf{w}|\mathcal{L}))$, where $\mathcal{H} \subset \mathcal{W}^N$ is the set of sample hypotheses and $p(\mathbf{w}|\mathcal{L}) > 0$ is the probability that $\mathbf{w} \in \mathcal{H}$ was indeed the sample. When $\mathbf{w} \in \mathcal{W}^N - \mathcal{H}$, we define $p(\mathbf{w}|\mathcal{L}) = 0$. 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.[1]

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

$$\tilde{p}(w|\mathcal{L}) = \sum_{\mathbf{w} \in \mathcal{H}} \hat{p}(w|\mathbf{w})\, p(\mathbf{w}|\mathcal{L}) \tag{4}$$

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

We choose $\tilde{p}$ as our estimator of $p(w)$ from $\mathcal{L}$ since we believe Eq. 4 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 $\tilde{p}$ under some restrictions on $\hat{p}$.

Suppose that $\hat{p}(w|\mathbf{w})$ only depends on the count $r$ of $w$ in $\mathbf{w}$. This is true for the maximum likelihood and Good's estimate. Then:

$$\begin{aligned} \tilde{p}(w|\mathcal{L}) &= \sum_{\mathbf{w} \in \mathcal{H}} \hat{p}(w|r)\, p(\mathbf{w}|\mathcal{L}), \qquad r = c(w, \mathbf{w}) \\ &= \sum_{\mathbf{r}=0}^{N} \hat{p}(w|r)\, p(r|w, \mathcal{L}) \end{aligned} \tag{5}$$

Thus, in these cases we can determine $\tilde{p}$ from $p(r|w, \mathcal{L})$.

For the maximum likelihood estimator, Eq. 5 further simplifies to:

$$\tilde{p}(w|\mathcal{L}) = \frac{1}{N} \sum_{\mathbf{r}=0}^{N} r\, p(r|w, \mathcal{L}) = \frac{\mathcal{E}[r|w, \mathcal{L}]}{N} \equiv \frac{\bar{r}}{N} \tag{6}$$

For Good's estimate, we need an estimate of $\mathcal{E}_N[n_r]$ to use Eq. 5. We can use:

$$\mathcal{E}_N[n_r] = \sum_{w \in \mathcal{W}} p_N(r|w) \approx \sum_{w \in \mathcal{W}} p(r|w, \mathcal{L}) \tag{7}$$

### 2.3. Properties of $\tilde{p}$:

Our estimator $\tilde{p}$ defined in Eq. 4 has the following properties:

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

[1] 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.

2. $\mathbf{Bias}(\tilde{p}) = \mathbf{Bias}(\hat{p})$:

$$\begin{aligned}\mathcal{E}[\tilde{p}(w)] &= \sum_{\mathcal{L}} \tilde{p}(w|\mathcal{L})\, p(\mathcal{L}) \\ &= \sum_{\mathcal{L}} \sum_{\mathbf{w}\in\mathcal{H}} \hat{p}(w|\mathbf{w})\, p(\mathbf{w}|\mathcal{L})\, p(\mathcal{L}) \\ &= \sum_{\mathbf{w}\in\mathcal{W}^N} \hat{p}(w|\mathbf{w}) \sum_{\mathcal{L}} p(\mathbf{w}|\mathcal{L})\, p(\mathcal{L}) \\ &= \sum_{\mathbf{w}\in\mathcal{W}^N} \hat{p}(w|\mathbf{w})\, p(\mathbf{w}) = \mathcal{E}[\hat{p}(w)] \quad (8)\end{aligned}$$

So $\tilde{p}$ has the same bias as $\hat{p}$ and, in particular, is unbiased if $\hat{p}$ is.

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

$$\begin{aligned}|\tilde{p}(w|\mathcal{L}) - p(w)| &= \left|\sum_{r=0}^{N} \hat{p}(w|r)\; p_N(r|w,\mathcal{L}) - p(w)\right| \\ &\le |\hat{p}(w|r_0)\; p_N(r_0|w,\mathcal{L}) - p(w)| + \\ &\quad \sum_{\substack{r=0 \\ r\neq r_0}}^{N} \hat{p}(w|r)\; p_N(r|w,\mathcal{L}) \\ &\le |\hat{p}(w|r_0)\; p_N(r_0|w,\mathcal{L}) - p(w)| \\ &\quad +1 - p_N(r_0|w,\mathcal{L}) \le 3\epsilon \quad (9)\end{aligned}$$

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, $\hat{p}_g$. When $\mathbf{w}$ is hidden, it is:

$$\begin{aligned}\mathcal{E}[p_u|c(u,\mathbf{w}) = c(w,\mathbf{w})] &= \\ \sum_{\mathbf{w}\in\mathcal{H}} \mathcal{E}[p_u|c(u,\mathbf{w}) = c(w,\mathbf{w}),\mathbf{w}]\; p(\mathbf{w}|\mathcal{L}) & \\ = \sum_{r=0}^{N} \mathcal{E}[p_u|r]\; p(r|w,\mathcal{L}) & \\ = \sum_{r=0}^{N} \hat{p}_g(w|r)\; p(r|w,\mathcal{L}) \quad (10)\end{aligned}$$

From Eq. 5 we see that Eq. 10 equals $\tilde{p}$ when $\hat{p}$ 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.

### 2.4. Computing $\tilde{p}$

We can compute the $\tilde{p}$ in Eq. 4 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 $p(\mathbf{w}|\mathcal{L})$. Then

$$\tilde{p}(w|\mathcal{L}) \approx \frac{1}{M}\sum_{i=1}^{M} \hat{p}(w|\mathbf{w}_i) \quad (11)$$

For more direct methods, we need to restrict $\hat{p}$. Let us require that it depends only on the count $r$ as in Eq. 5. In order to use Eq. 5, we need to evaluate:

$$p(r|w,\mathcal{L}) = \sum_{\substack{\mathbf{w}\in\mathcal{H}, \\ c(w,\mathbf{w})=r}} p(\mathbf{w}|\mathcal{L}) \quad (12)$$

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 $\mathcal{L}_i = (\mathcal{H}_i,\ p(\mathbf{w_i},\ \mathcal{L}_i))$, $\mathcal{H}_i \subset \mathcal{W}^{N_i}$, $N_1 + N_2 = N$, $\mathcal{H} = \mathcal{H}_1\mathcal{H}_2$, and $p(\mathbf{w_1 w_2}|\mathcal{L}) = p(\mathbf{w_1}|\mathcal{L}_2)\, p(\mathbf{w_2}|\mathcal{L}_2)$ for all $\mathbf{w_1}\in\mathcal{H}_1$ and $\mathbf{w_2}\in\mathcal{H}_2$. Then:

$$p(r|w,\mathcal{L}) = \sum_{k=0}^{r} p(r-k|w,\mathcal{L}_1)\, p(k|w,\mathcal{L}_2) \quad (13)$$

More generally, if $\mathcal{L}$ can be divided into $M$ mutually independent parts, then Eq. 13 can be iteratively applied $M-1$ times to determine $p(r|w,\mathcal{L})$. In the speech recognition example, $\mathcal{H}$ might correspond to a long passage while each $\mathcal{H}_i$ could be hypotheses for the $i$th sentence in the passage where we assume sentences are independent of each other.

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

## 3. UNSUPERVISED LM ADAPTATION

In this section, we will use the Monte Carlo approach of Eq. 11 with the Good-Turing estimate to generalize Katz backoff modeling in an adaptation scenario. To build an adapted n-gram model, we use a count merging approach, much as presented in [3], which is shown there to be a special case of maximum a posteriori (MAP) adaptation. Let $\mathbf{w}_0$ be the prior, out-of-domain corpus, and $\mathbf{w}_i$ be the $i$th randomly sampled corpus from the set of word lattices, $1 \le i \le M$. Let $h$ represent an n-gram history of zero or

| Adaptation Approach | Samples | Hours of unlabeled data | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 0.6 | 1.9 | 3.7 | 7.4 | 14.3 | 28.3 | 56.8 | 114.4 | 229.4 | 456.9 | 1047.9 |
| One-best | 1 | 49.1 | 49.3 | 49.8 | 50.3 | 50.5 | 50.9 | 51.3 | 51.6 | 52.0 | 52.4 | 52.6 | 52.8 |
| Lattices | 1000 | 49.1 | 49.4 | 50.0 | 50.5 | 50.8 | 51.2 | 51.7 | 52.2 | 52.4 | 52.5 | 52.8 | 52.8 |

**Table 1**. Word accuracy for varying amounts of unlabeled adaptation data for both adaptation setups

more words. Let $c_i(hw)$ denote the raw count of an n-gram $hw$ in $\mathbf{w}_i$. Let $\hat{p}_i(hw)$ denote the standard Katz backoff model estimate of $hw$ given $\mathbf{w}_i$. We define the corrected count of an n-gram $hw$ as:

$$\hat{c}_i(hw) \quad = \quad |\mathbf{w}_i|\,\hat{p}_i(hw) \tag{14}$$

Then:

$$\tilde{p}(w \mid h) = \frac{\tau_h \hat{c}_0(hw) + \frac{1}{M}\sum_1^M \hat{c}_i(hw)}{\sum_{w'}\{\tau_h \hat{c}_0(hw') + \frac{1}{M}\sum_1^M \hat{c}_i(hw')\}} \tag{15}$$

where $\tau_h$ is a state dependent parameter that dictates how much the out-of-domain prior counts should be relied upon. Note that the in-domain contribution to the above formula is taken directly from the Monte Carlo estimate in Eq. 11. The model is then defined as:

$$p^*(w \mid h) = \begin{cases} \tilde{p}(w \mid h) & \text{if } c_i(hw) > 0 \text{ for some } i \\ \alpha\, p^*(w \mid h') & \text{otherwise} \end{cases} \tag{16}$$

where $\alpha$ is the backoff weight and $h'$ the backoff history for history $h$.

The principal difficulty in MAP adaptation of this sort is determining the mixing parameters $\tau_h$ in Eq. 15. Following [3], we chose a single mixing parameter for each model that we built, i.e. $\tau_h = \tau$ for all states $h$ in the model.

Unsupervised MAP adaptation, however, differs from supervised MAP adaptation in one respect, which becomes critical for dealing with the large amount of unlabeled data in our experiments. 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 found that the empirically optimized mixing parameters were very different for different unlabeled sample sizes. To simplify the parameter estimation process across all conditions, we made the simplifying assumption that $\tau = \tau_0 \frac{\sum_{i=1}^M |\mathbf{w}_i|}{M|\mathbf{w}_0|}$ 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, $\tau_0 = 3.5$. This resulted in good performance for both word-lattice sampling and one-best adaptation approaches.

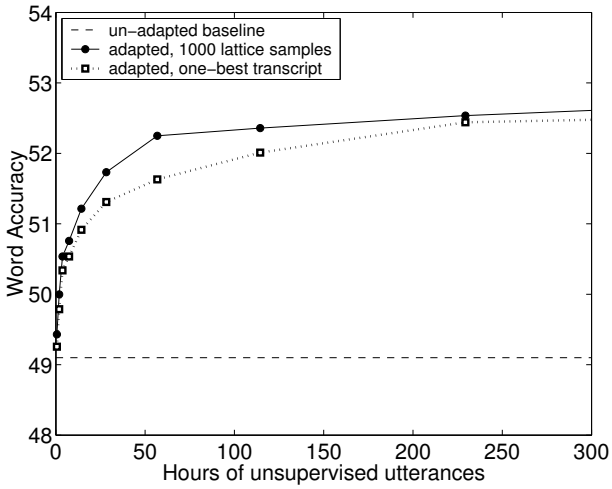


**Fig. 1**. Word Accuracy versus hours of unlabeled utterances for adaptation based on word lattice sampling and ASR one-best output, for up to 300 hours of in-domain utterances.

## 4. 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 dialogue application known as Customer Care (CC) [9]. The out-of-domain corpus was 171,343 words transcribed from a previously deployed application known as Operator Services (OS) [9]. 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, and adapt the OS baseline model with counts derived from the ASR output, as outlined in the previous section[2].

[2]The one-best transcription is considered one sample for equation 15.

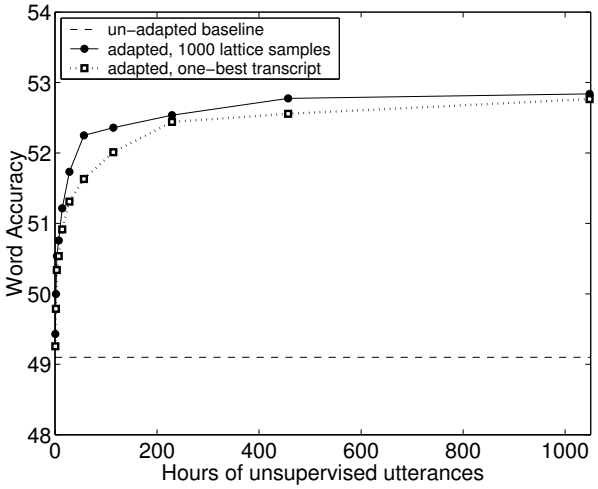


**Fig. 2**. Word Accuracy versus hours of unlabeled utterances for adaptation based on word lattice sampling and ASR one-best output, for over 1000 hours of in-domain utterances. Note that this is the same plot as in figure 1, with a different range.

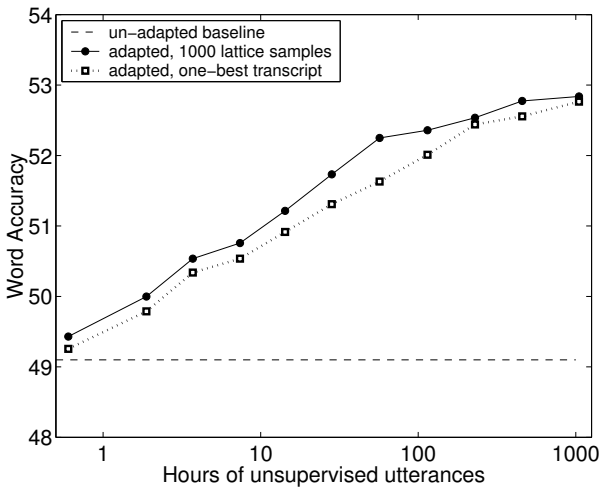


**Fig. 3**. Word Accuracy versus hours of unlabeled utterances for adaptation based on word lattice sampling and ASR one-best output, with the hours plotted in log scale.

Table 1 presents the 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 affect. 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 point 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.

Figure 1 plots these results versus the baseline from zero to 300 hours of in-domain unlabeled data. Figure 2 gives a wider view including all of the trials, and figure 3 shows these results with the hours of unlabeled data in log scale.

Table 2 presents perplexity results on the test corpus for the same breakdown of trials as table 1. Out-of-vocabulary words (3.5 percent of tokens in the test set) were mapped to an unknown token, which was reserved a unigram probability of 0.00001. Figure 4 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 sampling approach in terms of perplexity as well as word accuracy. Figure 5 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.

## 5. DISCUSSION

The results presented in the last section demonstrate that one-best transcriptions are effective for unsupervised MAP adaptation, and can provide several points of improvement, even when the baseline accuracy is poor. Our results here are consistent with those presented in [3]. The estimation is simple when using one-best transcriptions, so this is likely to be the method of choice for many doing adaptation, particularly 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, there is a small, consistent gain to be had by using the word lattices, when the amount of unlabeled data is limited. This makes our approach more attractive in scenarios when the data is limited or highly fragmented – e.g., for self-adaptation.

Future research is threefold. First, we intend to investigate an efficient algorithm for doing Good-Turing estimation exactly, without requiring the Monte Carlo technique used for this paper. Second, we intend to investigate this method in other domains, where the baseline accuracies and quality of the posterior probability estimates are somewhat better than what is to be had in this domain. Improved baseline accuracies will make the one-best approach more effective (in the limit as effective as supervised adaptation), but the improved posteriors will also have an effect on lattice-based methods, leading to better estimates.

Finally, we would like to have a better understanding of what exactly is being learned from the unlabeled data. Are a small number of frequent collocations receiving much higher probabilities, or is the overall shape of the resulting distribution driving the improvements? If it is the former, then perhaps there are techniques for extracting a relatively small number of surprisingly likely collocations and simply moving the distribution to accommodate their increased likelihood. In other words, perhaps there are more effective

| Adaptation Approach | Samples | Hours of unlabeled data | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 0 | 0.6 | 1.9 | 3.7 | 7.4 | 14.3 | 28.3 | 56.8 | 114.4 | 229.4 | 456.9 | 1047.9 |
| One-best | 1 | 179.7 | 133.4 | 123.8 | 117.4 | 112.6 | 107.3 | 102.0 | 95.6 | 90.4 | 84.8 | 80.5 | 76.0 |
| Lattices | 1000 | 179.7 | 131.0 | 118.2 | 109.2 | 102.4 | 96.2 | 90.0 | 83.7 | 79.8 | 76.4 | 74.4 | 72.4 |

**Table 2**. Perplexity of the test set for models trained on varying amounts of unlabeled adaptation data for both adaptation setups. OOV words were mapped to token ⟨unknown⟩, which was given an $\epsilon$ probability.

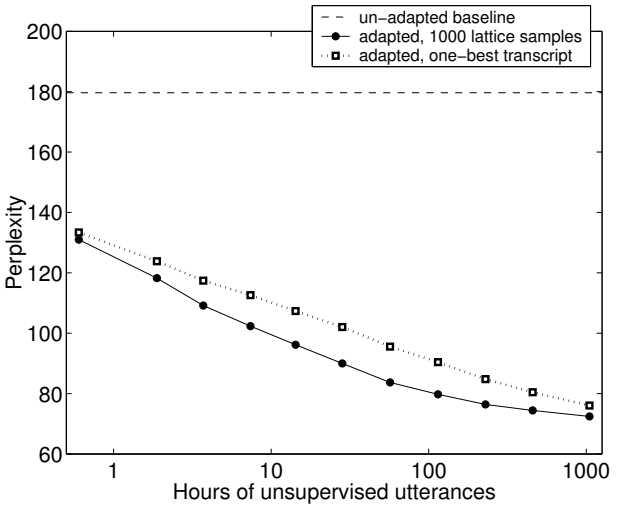


**Fig. 4**. Perplexity versus hours of unlabeled utterances for adaptation based on word lattice sampling and ASR one-best output, with the hours plotted in log scale.

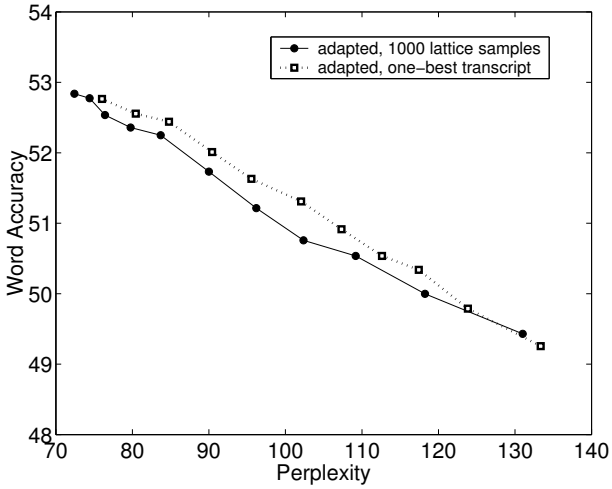


**Fig. 5**. Word Accuracy versus Perplexity for adaptation based on word lattice sampling and ASR one-best output.

adaptation techniques for making use of noisy ASR output. In any case, this paper has presented a method for estimating corrected counts for n-grams while using the distribution over hypotheses provided by the word lattice.

## 6. REFERENCES


[1] Roberto Gretter and Giuseppe 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)*, 2001, pp. 557–560.

[2] Andreas Stolcke, "Error modeling and unsupervised language modeling," in *Proceedings of the 2001 NIST Large Vocabulary Conversational Speech Recognition Workshop*, Linthicum, Maryland, May 2001.

[3] Michiel Bacchiani and Brian Roark, "Unsupervised language model adaptation," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003, pp. 224–227.

[4] Frederick Jelinek and Robert L. Mercer, "Interpolated estimation of markov source parameters from sparse data," in *Proceedings of the Workshop on Pattern Recognition in Practice*, 1980, pp. 381–397.

[5] Ian H. Witten and Timothy C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE Transactions on Information Theory*, vol. 37, no. 4, pp. 1085–1094, 1991.

[6] Slava 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*, vol. 35, no. 3, pp. 400–401, 1987.

[7] Reinhard Kneser and Hermann Ney, "Improved backing-off for m-gram language modeling," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1995, pp. 181–184.

[8] Irving J. Good, "The population frequencies of species and the estimation of population parameters," *Biometrica V*, vol. 40, no. 3,4, pp. 237–264, 1953.

[9] Allen L. Gorin, Alicia Abella, Tirso Alonso, Giuseppe Riccardi, and Jeremy H. Wright, "Automated natural spoken dialogue," *IEEE Computer Magazine*, vol. 35, no. 4, pp. 51–56, 2002.