Language Model Adaptation through Shared Linear Transformations

Wei-Yun Ma [1], *Yun-Cheng Ju* [2], *Xiaodong He* [2], *Li Deng* [2]

[1] Columbia University, New York, NY 10027, USA
[2] Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA
ma@cs.columbia.edu, {yuncj,xiaohe,deng}@microsoft.com

June 2014

Technical Report

Microsoft Research
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Abstract

Language model (LM) adaptation is an active area in natural language processing and has been successfully applied to speech recognition and to many other applications. To provide fine-grained probability adaptation for each n-grams, we in this work propose three adaptation methods based on shared linear transformations: n-gram-based linear regression, interpolation, and direct estimation. Further, in order to address the problem of data sparseness, n-grams are clustered and those in the same cluster group are made to share the same adaptation parameters. We carry out evaluation experiments on a domain adaptation task with limited adaptation data. The experimental results show that the best LM after our adaption method can reduce the perplexity by half compared with the baseline LM without adaptation, and that it also achieves a perplexity reduction of 15% compared with the earlier state-of-the-art LM adaptation methods. The speech recognition experimental results show that the proposed LM adaptation method reduces the WER by 20.8% compared with the baseline LM without adaptation.

Index Terms: Language model adaptation, linear regression, interpolation, direct estimation, clustering

1. Introduction

Language model (LM) adaptation is an active area in natural language processing and has been successfully applied to speech recognition and to many other applications [3][5][10][13][14][16]. It attempts to adjust the parameters of a LM trained from a general domain using a large amount of data (i.e., the background corpus) so that the adjusted LM can perform well on a particular domain, for which only a small amount of data (i.e., the adaptation corpus) is available.

1.1. Existing techniques

The adaptation techniques can be grouped into two categories: linear interpolation [11][4][17][21][2][18][1][22][26] and constraint specification [8][12][24]. Among the many linear interpolation approaches, according to different levels on which the interpolation is applied, they can be further divided into model-level, count-level, and topic-level linear interpolation.

Linear interpolations at the model and topic levels are rather straightforward. Take the former as an example: It first estimates the LM probabilities derived from the Background corpus and the Adaptation corpus, respectively, and then combines the two probabilities through linear interpolation. In order to provide a more fine-grained adaptation, in [4], Bacchiani et al suggested a linear interpolation framework based on the maximum a posteriori (MAP) estimation to adapt LM. This leads to a solution of linear interpolation at the frequency count level rather than at the model level:

$$P^{CM}(w|h) = \frac{C^B(hw)+\lambda C^A(hw)}{C^B(h)+\lambda C^A(h)} \qquad (1)$$

where w denotes the word, h denotes the n-gram history of w, $C^B(h)$ and $C^B(hw)$ are the count of h and hw in the background dataset, $C^A(h)$ and $C^A(hw)$ are the count of h and hw in the adaptation dataset, λ is a constant factor and can be estimated empirically. Bacchiani et al. reported this MAP based LM adaptation method gives superior performance.

1.2. Motivations of this work

Let us consider the parameters in these linear interpolation-based approaches. Here, the n-grams of the same domain/topic or word history share the same interpolation weights. Therefore, it is difficult to carry out fine-grained adaption of LM probabilities for each n-gram or n-gram type. On the other hand, each n-gram/n-gram type is supposed to have its own behavior under different domains, and therefore, n-gram specific parameter adaptation is needed. In order to address this problem, we propose three new LM adaptation methods: n-gram-based linear regression, interpolation, and direct estimation. In the n-gram-based linear regression method, we scale and shift each n-gram probability at the log-probability domain based on constrained optimization toward cross entropy on the adaptation corpus. We also investigate scaling of each n-gram probability at the original probability domain. In the second method, for n-gram-based interpolation, all n-grams in a same n-gram group share the same interpolation weight. In the third method, we directly re-estimate the LM on the adaptation corpus, followed by a normalization process to adjust each n-gram probability.

In order to avoid overfitting and to be able to adapt the LM using extremely limited adaptation data, we cluster n-grams into several groups and estimate for each n-gram group a set of shared adaptation parameters so that the information from observed n-grams can be propagated to unseen n-grams within the same group during adaptation.

2. N-gram Clustering

With many individual n-gram probabilities to be adapted while having a small amount of adaptation data available, we face the problem that many n-grams are not observed in the adaptation corpus. For those n-grams that have not been observed in the adaptation data, it is desirable to exploit the observed n-grams similar to them to help adapt their n-gram probabilities. Thus, we propose to group similar n-grams into a cluster and to make them share the same set of adaption parameters. By doing so, our models are able to perform reliable parameter adaptation given often an extremely limited amount of adaptation data in practical applications.

We first build word clusters through the Brown word clustering algorithm [7], which is a form of hierarchical clustering of words based on the contexts in which they occur. We then perform n-gram grouping based on the word clusters. For example, given two n-grams, the *x*-th n-gram ($w_1..w_I$) and the y-th n-gram ($u_1..u_I$). We designed two criteria to decide if the two n-grams belong to the same n-gram class as follows. Criteria 1 is for n-gram-based linear regression and interpolation models, while Criteria 2 is for direct estimation model.

Criteria 1: g($w_1..w_I$)=g($u_1..u_I$)
when C(w_I)=C(u_I) and h_x=h_y , i.e, $w_1..w_{I-1} = u_1..u_{I-1}$

Criteria 2: g($w_1..w_I$)=g($u_1..u_I$)
when C(w_1)=C(u_1), C(w_2)=C(u_2),…, and C(w_I)=C(u_I)

where C(.) is the Brown word classifying function. h_* is the word history of the *th n-gram. g(.) is the n-gram grouping function of n-gram index using Criteria 1, and n-gram using Criteria 2, accordingly.

3. N-gram-based Linear Regression

Inspired by the successful linear regression-based model adaptation methods in speech recognition [14, 18, 19] and in web search [24], we propose a novel linear regression framework to perform LM adaptation. For each n-gram, its adapted LM probability is obtained by applying scaling and shifting to the log probability of background LM. We also investigate scaling at the original probability domain.

3.1. Linear regression on N-gram log-probabilities

Let $P_k^B(x_k|h_k)$ be LM probability of the k-th n-gram trained on the background corpus and $P_k(x_k|h_k)$ be its adapted N-gram probability. $ln\big(P_k(x_k|h_k)\big)$ can be estimated by linear operations of scaling and shifting:

$$ln\big(P_k(x_k|h_k)\big) = a_{g(k)} ln(P_k^B(x_k|h_k)) + b_{g(k)} \qquad (2)$$

where x_k and h_k are the word and word history of the k-th n-gram LM probability; $g(k)$ is a n-gram group function, which maps the k-th n-gram to the cluster that it belongs to. $a_{g(k)}$ and $b_{g(k)}$ denote the scaling and shifting operations applied to $ln(P_k^B(x_k|h_k))$.

We use cross entropy toward the adaptation corpus plus regularization term as our object function for adaptation. Our goal is to obtain a set of $a_{g(k)}$ and $b_{g(k)}$ so that the object function f_0 is minimized.

$$f_0 \equiv -\sum_{k=1}^{D}(count_k \times ln\big(P_k(x_k|h_k)\big) + \mathrm{R}(a_{g(k)}, b_{g(k)})) \qquad (3)$$

$$\mathrm{R}(a_{g(k)}, b_{g(k)})) \equiv \sum_{k=1}^{D} \lambda_a\big(a_{g(k)} - 1\big)^2 + \lambda_b {b_{g(k)}}^2 \qquad (4)$$

where R(.) is the regularization function, D is the total number of n-grams occurring in the adaptation corpus, and $count_k$ is the count of the *k*-th n-gram occurring in the adaptation corpus. λ_a and λ_b are two weighting factors of the regularization term, which are tuned on a held-out data set in our implementation.

The adapted n-gram probabilities also need to satisfy the constraint that the sum of probabilities of the same word history is less than a threshold which is less than one. The reason of not restricting the sum to be exactly one is that we need to reserve certain probability mass for the LM back-off.

Constraint 1: $\sum_{\{j|h_j=h\}} P_j(x_j|h_j) \leq threshold_h$

where $threshold_h$ is a value between 0 and 1. In our practice, we decide this value as $\sum_{\{j|h_j=h\}} P_j^{LI}(x_j|h_j)$, where $P_j^{LI}(x_j|h_j)$ is the LM probabilities obtained from linear interpolation at the model level. Constraint 1 is equivalent to the constraint function:

$$CF_1\ (\mathrm{h}) \equiv \sum_{\{j|h_j=h\}} P_j(x_j|h_j) \text{ - } threshold_h \leq 0 \qquad (5)$$

Since (2) is equivalent to adjusting the exponent of n-gram probability

$$P_k(x_k|h_k) = P_k^B(x_k|h_k)^{\,a_{g(k)}} \times e^{b_{g(k)}} \qquad (6)$$

we can observe that $P_k(x_k|h_k)$ is always larger than 0, so there is no need to check if it is negative during optimization, which simplifies the optimization.

Our object and constrain functions are convex functions, so we can use any constrained convex optimization algorithm to obtain $a_{g(k)}$ and $b_{g(k)}$. In implementation, we use the inequality-constrained subgradient method [6] to serve this purpose. The algorithm is described in Figure1.

```
Begin while loop
  If all constrains are satisfied, then for all n-gram group g, do
    $a_g \leftarrow a_g - \eta \frac{\partial f_0}{\partial a_g}$
    $= a_g - \eta \sum_{\{k|g(k)=g\}}(-count_k \times \ln(P_k^B(x_k|h_k) + \lambda_a(2a_{g(k)}^t - 2))$
    $b_g \leftarrow b_g - \eta \frac{\partial f_0}{\partial b_g} = b_g - \eta \sum_{\{k|g(k)=g\}}(-count_k + 2\lambda_b b_{g(k)})$
  If Constraint 1 is violated for the word history h, then for all n-gram group g, do
    $a_g \leftarrow a_g - \eta \frac{\partial CF_1(h)}{\partial a_g}$
    $= a_g - \eta \sum_{\{k|h_k=h,g(k)=g\}}(P_k^B(x_k|h_k)^{a_{g(k)}} \times \ln(P_k^B(x_k|h_k) \times e^{b_{g(k)}})$
    $b_g \leftarrow b_g - \eta \frac{\partial CF_1(h)}{\partial b_g}$
    $= b_g - \eta \sum_{\{k|h_k=h,g(k)=g\}} P_k^B(x_k|h_k)^{\,a_{g(k)}} \times e^{b_{g(k)}}$
End while loop
```

Figure 1. Algorithm of Linear Regression on N-gram Log Probabilities

3.2. Linear Regression on N-gram probabilities

We also investigate linear regression based adaptation on n-gram probabilities, i.e.

$$P_k(x_k|h_k) = a_{g(k)} P_k^B(x_k|h_k) \qquad (7)$$

We use cross-entropy on the adaptation corpus plus regularization term as our object function for optimization, which is defined in (8) below:

$$f_0 \equiv -\sum_{k=1}^{D}(count_k \times ln\big(P_k(x_k|h_k)\big) + \mathrm{R}(a_{g(k)}) \qquad (8)$$

$$\mathrm{R}(a_{g(k)})) \equiv \sum_{k=1}^{D} \lambda_a (a_{g(k)} - 1)^2 \qquad (9)$$

In addition to Constraint 1, we need to assure every n-gram probability be equal or larger than zero. In practice, we assure that by checking the scaling value, i.e.

Constraint 2: $0 < \varepsilon \le a_g$

In our implementation, ε is set to 0.1 to prevent the probabilities of the n-grams which do not appear in the adaptation corpus from becoming zero during optimization.

Using the subgradient method, we can get the algorithm shown in Figure2.

```
Begin while loop
  If all constrains are satisfied, then for all n-gram group g, do
    $a_g \leftarrow a_g - \eta \frac{\partial f_0}{\partial a_g}$
    $= a_g - \eta \sum_{\{k|g(k)=g\}}(\frac{-count_k}{a_{g(k)}} + \lambda_a(2a_{g(k)} - 2))$
  If Constraint 1 is violated for the word history h, then for all n-gram group g, do
    $a_g \leftarrow a_g - \eta \frac{\partial CF_1(h)}{\partial a_g} = a_g - \eta \sum_{\{k|h_k=h,g(k)=g\}} P_k^B(x_k|h_k)$
  If Constraint 2 is violated for any n-gram group g
    $a_g \leftarrow 0.1$
End while loop
```

Figure 2. Algorithm of Linear Regression on N-gram Probabilities

4. N-gram-based Linear Interpolation

Linear interpolation at the n-gram level is pretty straightforward. We combine $P_k^B(x_k|h_k)$ and $P_k^A(x_k|h_k)$ through the following linear interpolation:

$$P_k(x_k|h_k) = \alpha_{g(k)} P_k^B(x_k|h_k) + (1 - \alpha_{g(k)}) P_k^A(x_k|h_k) \qquad (10)$$

where $0 \le \alpha_{g(k)} \le 1$ serve as the interpolation coefficient. The object function is defined as (11).

$$f_0 \equiv -\sum_{k=1}^{D}(count_k \times ln\big(P_k(x_k|h_k)\big)) \qquad (11)$$

```
Begin while loop
  If all constrains are satisfied, then for all n-gram group g, do
    $\alpha_g \leftarrow \alpha_g - \eta \frac{\partial f_0}{\partial \alpha_g} =$
    $\alpha_g - \eta \sum_{\{k|g(k)=g\}} (\frac{-count_k \times (P_k^B(x_k|h_k) - P_k^A(x_k|h_k))}{\alpha_g \times \left(P_k^B(x_k|h_k) - P_k^A(x_k|h_k)\right) + P_k^A(x_k|h_k)})$
  If Constraint 1 is violated for the word history h, then for all n-gram group g, do
    $\alpha_g \leftarrow \alpha_g - \eta \frac{\partial CF_1(h)}{\partial \alpha_g}$
    $= \alpha_g - \eta \sum_{\{k|h_k=h,g(k)=g\}} (P_k^B(x_k|h_k) - P_k^A(x_k|h_k))$
End while loop
```

Figure 3. Algorithm of N-gram-based Linear Interpolation

The optimized target corpus now is a held-out corpus instead of the adaptation corpus. Our goal is to obtain a set of $\alpha_{g(k)}$ so that the object function f_0 is minimized under the constraint 1. Using the subgradient method, we can get the algorithm shown in Figure 3.

5. N-gram-based Direct Estimation

The algorithm of n-gram-based direct estimation is described formally in Figure 4. I.e., for each n-gram cluster, we first estimate the n-gram cluster probabilities in both the background domain and the adaptation domain, as shown in step 1 of Figure 4. Then, we obtain a temporarily adapted LM probability for each n-gram according to step 2 of Figure 4. At the last step, we normalize the temporarily adapted LM probability toward the word history and obtained the final LM probability for each n-gram.

Step1: obtain n-gram group probabilities- $P^B(G_u|G_v)$ and $P^A(G_u|G_v)$ by

$$P^B(G_u|G_v) \equiv \frac{\sum_{\{w,h|g(w)=G_u,g(h)=G_v\}} P^B(h,w)}{\sum_{\{h|g(h)=G_v\}} P^B(h)}$$

$$P^A(G_u|G_v) \equiv \frac{\sum_{\{w,h|g(w)=G_u,g(h)=G_v\}} P^A(h,w)}{\sum_{\{h|g(h)=G_v\}} P^A(h)}$$

Step2: obtain temporarily adapted LM probability by multiplying background LM probability with the ratio of N-gram group probability:

$$P^T(w|h) = P^B(w|h) \times \frac{P^A(g(w)|g(h))}{P^B(g(w)|g(h))}$$

Step3: normalize the temporarily adapted LM probability toward the word history:

$$P(w|h) = \frac{P^T(w|h)}{\sum_{\{x\}} P^T(x|h)}$$

Figure 4. Algorithm of N-gram-based Direct Estimation

6. Experiments

In order to evaluate the proposed adaptation models, we carried out a series of experiments under a rather difficult environment that only very limited adaptation data are available. Perplexity and word error rate

(WER) of our in-house automatic speech recognition (ASR) system are the metrics used for performance evaluation.

6.1. Datasets and experimental setting

We used a part of Gigaword 4.0 as our background corpus, which contains all newswires of New York Time in 2008. We extracted part of a MS Excel-related book of 2010 as our adaptation corpus and held-out corpus. We also used another MS Excel-related book of 2013 as the testing corpus. The sizes of the data sets in our experiments are summarized in Table 1.

Table 1. Summary of datasets

Data sets	Sentence s	Words
Background corpus (bg)	1528343	69674766
Adaptation corpus (adapt)	120	3948
Held-out corpus	120	4390
Testing corpus	4597	111219

We built the background LM and the adaptation LM from background corpus and adaptation corpus, respectively, using modified Kneser-Ney smoothing [9] provided by SRILM [23].

6.2. General results

The evaluation results are presented in Table 2. First, we observe that the proposed N-gram-level fine-grained LM adaptation methods achieve significant perplexity and WER reduction comparing to using the background LM without adaptation (LM of bg). For example, the N-gram-based linear interpolation (N-gram Int) approach reduces the perplexity by half, from 756 to 373, and reduces the WER from 16.40% to 12.99%, corresponding to a 20.8% relative WER reduction. This demonstrates the effectiveness of the proposed LM adaptation methods.

Table 2. Performance on held-out and testing corpuses. (The number of word clusters is set to 200)

	PPL on held-out set	PPL on test set	ASR WER (%)
LM of bg (baseline)	812	756	16.40
LM of adapt (baseline)	795	845	22.76
MAP adaptation (baseline)	394	430	13.35
LM of bg+adapt (baseline)	711	724	-
Class-based LM (baseline)[1]	782	846	-
N-gram Reg on prob	360	370	13.23
N-gram Reg on log prob	361	**365**	13.24
N-gram Int	302	373	**12.99**
N-gram DirEst	428	453	13.34

In order to further evaluate the effectiveness of the proposed methods compared to previous work, we implemented three LM adaptation baselines for comparison. They are based on MAP-based LM adaptation (MAP adaptation), multi-style training, e.g., re-estimate the LM on the combination of background data and adaptation data (LM of bg+adapt), and Class-based LM. Compared to these baselines, all the proposed methods except N-gram based direct estimation (n-gram DirEst) give superior performance in both perplexity and WER. Specifically, compared with the strong baseline which is based on MAP LM

[1] Our class-based LM consists of two parts: the probability of word given class, which is learned from bg+adapt corpus, and N-gram of word classes, which is learned from ada. Use of more advanced class-based LMs, such as model M [8][22] will be our future work.

adaptation, the linear regression on log probability method (N-gram Reg on log prob) achieves a perplexity reduction of 65 (from 430 to 365, a 15% relative perplexity reduction). The N-gram-based linear interpolation (N-gram Int) also gives significant perplexity reductions (from 430 to 373, a 13.2% perplexity reduction) compared to the MAP based LM adaptation baseline. These results demonstrated that the newly proposed methods can adapt the LM using limited adaptation data more effectively.

On the other hand, the N-gram based direct estimation (N-gram DirEst), though outperforming the two methods without adaptation, gives a moderate performance compared with previous LM adaptation baselines. Our interpretation of the moderate performance is that it is due to the normalization step: since the normalization does not consider the adaptation corpus, it could weaken the effect of the adaptation operations of step 2 of Figure 4. Therefore, how to design an improved normalization for "N-gram DirEst" is in our future work.

6.3. Effects of the size of N-gram classes

To further investigate the effects of n-gram clustering, we compare the effects of different sizes of n-gram classes. As described in section 2, our n-gram grouping is based on Brown word clustering. Thus, we can control the number of n-gram classes by controlling the number of word clusters. Their sizes are summarized in Table 3.

Table 3. The relationship between the word cluster number and the n-gram class number using Criteria 1

word cluster #	unigram class #	bigram class #	trigram class #
206K (no clustering)	206K (no grouping)	2369K (no grouping)	3480K (no grouping)
2000	2002	1533K	3096K
500	502	1177K	2764K
200	202	954K	2512K
20	22	461K	1791K

Table 4. The effect of different numbers of word clusters (numbers of n-gram classes) on the performance of linear regression on N-gram log probability

word cluster #	Perplexity on held-out corpus	Perplexity on testing corpus	ASR WER (%)
206K	373	381	13.42
2000	368	362	13.21
500	366	371	13.78
200	361	365	13.24
20	369	376	13.21

The effects of different numbers of word clusters (numbers of n-gram classes) on the performance of Regression on N-gram Log Probability are shown in Table 4. We observe that 200 word clusters and 2000 word clusters give the best performance on held-out corpus and testing corpus, respectively. Therefore, by comparing them with the performance of no word clustering, it can be concluded that sharing the same adaptation parameters among n-grams that belong to a same cluster brings solid benefits in our adaptation algorithms.

7. Conclusions

In this paper, we propose and evaluate three new LM adaptation methods: 1) n-gram-based linear regression on log-probabilities and on linear probabilities, 2) interpolation at the n-gram level, and 3) direct estimation of adapted n-gram probabilities. These methods aim to provide fine-grained n-gram probability adaptation for a language model under the condition of small amounts of adaptation data. To avoid overfitting and address the sparseness problem, n-grams are grouped and the same set of adaptation parameters are shared among all n-grams in the same cluster. We carried out a series of experiments on a LM domain adaptation task. Experimental results show that all three proposed models achieve significant improvement in perplexity and WER over a strong baseline system, and that n-gram-based linear regression on log probabilities gives the lowest perplexity among all.

8. References

[1] G. Adda, M. Jardino, and J. Gauvain. “Language modeling for broadcast news transcription,” Proc. Eurospeech, pp. 1759–1762, 1999.

[2] C. Allauzen and M. Riley, “Bayesian language model interpolation for mobile speech input,” Proc. Interspeech, pp. 1429–1432, 2001.

[3] A. Axelrod, X. He, and J. Gao. “Domain adaptation via pseudo in-domain data selection,” Proc. EMNLP, 2011.

[4] M. Bacchiani, M. Riley, B. Roark, and R. Sproat, “MAP adaptation of stochastic grammars,” Computer Speech & Language, vol. 20, no. 1, pp. 41–68, 2006.

[5] J. R. Bellegarda, “Statistical language model adaptation: Review and perspectives,” Speech Communication Special Issue on Adaptation Methods for Speech Recognition, vol. 42, pp. 93–108, 2004.

[6] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[7] P. Brown, Peter Desouza, Robert Mercer, Vincent Pietra, and Jenifer Lai. “Class-based n-gram models of natural language,” Computational linguistics, 18(3):467–479, 1992.

[8] S. Chen, “Shrinking exponential language models,” Proc. NAACL-HLT, 2009.

[9] S. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” in Technical Report TR-10-98. Computer Science Group, Harvard University, 1998.

[10] L. Deng and D. O'Shaughnessy. Speech Processing --- A Dynamic and Optimization-Oriented Approach, Marcel Dekker Inc., June 2003.

[11] M. Federico and de Mori, R., “Language Model Adaptation” In: Ponting, K. (Ed.), Springer-Verlag, New York, 1999.

[12] M. Federico. “Efficient language model adaptation through MDI estimation,” Proc. Eurospeech, pp. 1583–1586, 1999.

[13] Q. Fu, X. He, and L. Deng. “Phone-Discriminating Minimum Classification Error (P-MCE) Training for Phonetic Recognition,” in Proc. Interspeech, August 2007.

[14] J. Gao, Hisami Suzuki, Wei Yuan. “An Empirical Study on Language Model Adaptation,” ACM Trans on Asian Language Information Processing, 5(2), pp. 207-227, 2006.

[15] X. He and W. Chou. “Minimum classification error linear regression for acoustic model adaptation of continuous density HMMs,” Proc. ICME, pp. I–397, 2003.

[16] X. He and L. Deng. “Speech-Centric Information Processing: An Optimization-Oriented Approach,” *Proceedings of the IEEE*, Vol. 31, 2013.

[17] B-J. Hsu, “Generalized Linear Interpolation of Language Models”, Proc. ASRU, 136–140, 2007.

[18] R. Iyer and M. Ostendorf. “Modeling Long Distance Dependence in Language: Topic Mixtures vs. Dynamic Cache Models,” Proc. ICSLP, 1996.

[19] C. Leggetter and P. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. Computer speech and language, 9(1):171, 1995.

[20] X. Lei, J. Hamaker, and X. He, “robust feature space adaptation for telephony speech recognition,” Proc. Interspeech, 2006.

[21] X. Liu, M.J.F. Gales, P.C. Woodland, Use of contexts in language model interpolation and adaptation, Computer Speech & Language, Volume 27, pp. 301-321, 2013.

[22] Sethy, S. Chen, E. Arisoy, B Ramabhadran, K. Audkhasi, S. Narayanan, P. Vozila. “Joint training of interpolated exponential n-gram models,” Proc. ASRU 2013.

[23] Stolcke. “SRILM – An extensible language modeling toolkit,” Proc. ICSLP, 2002.

[24] Y. C. Tam and T. Schultz, "Unsupervised Language Model Adaptation using Latent Semantic Marginals," Proc. Interspeech, 2006.

[25] H. Wang, X. He, M. Chang, Y. Song, R. White, and W. Chu. "Personalized Ranking Model Adaptation for Web Search," Proc. SIGIR, 2013.

[26] Y. Zhang, L. Deng, X. He, and A. Acero. "A Novel Decision Function and the Associated Decision-Feedback Learning for Speech Translation," in *ICASSP*, IEEE, 2011