Scaling Shrinkage-Based Language Models

Stanley F. Chen, Lidia Mangu, Bhuvana Ramabhadran, Ruhi Sarikaya, Abhinav Sethy

IBM T.J. Watson Research Center
P.O. Box 218, Yorktown Heights, NY 10598 USA
{stanchen,mangu,bhuvana,sarikaya,asethy}@us.ibm.com

***Abstract*—In [1], we show that a novel class-based language model, Model M, and the method of regularized minimum discrimination information (rMDI) models outperform comparable methods on moderate amounts of Wall Street Journal data. Both of these methods are motivated by the observation that *shrinking* the sum of parameter magnitudes in an exponential language model tends to improve performance [2]. In this paper, we investigate whether these shrinkage-based techniques also perform well on larger training sets and on other domains. First, we explain why good performance on large data sets is uncertain, by showing that gains relative to a baseline n-gram model tend to decrease as training set size increases. Next, we evaluate several methods for data/model combination with Model M and rMDI models on limited-scale domains, to uncover which techniques should work best on large domains. Finally, we apply these methods on a variety of medium-to-large-scale domains covering several languages, and show that Model M consistently provides significant gains over existing language models for state-of-the-art systems in both speech recognition and machine translation.**

I. Introduction

In [1], we proposed a novel class-based language model, *Model M*, that outperforms a Katz-smoothed word trigram model by 28% in perplexity and 1.9% absolute in automatic speech recognition (ASR) word-error rate; these are among the best results ever reported for a class-based language model. In addition, we showed that for the task of domain adaptation, the method of *regularized minimum discrimination information* (rMDI) modeling outperforms linear interpolation by up to 0.7% absolute in word-error rate (WER). However, these experiments were restricted to Wall Street Journal data with training sets less than 25 million words in length and were conducted with a non-state-of-the-art acoustic model. While Wall Street Journal is the canonical test bed for language modeling (LM) research, it is not representative of the data used in modern language modeling applications, many of which use languages other than English.

In this paper, we investigate whether the gains of Model M and regularized minimum discrimination information models scale to larger data sets, other domains and languages, and other applications, specifically, machine translation (MT). One particular concern is that both Model M and rMDI models were motivated as ways to *shrink* a word n-gram model. That is, when training and test data are drawn from the same distribution, it has been found for many types of exponential language models that

$$\log \text{PP}_{\text{test}} \approx \log \text{PP}_{\text{train}} + \frac{\gamma}{D} \sum_i |\tilde{\lambda}_i| \tag{1}$$

where PP_{test} and PP_{train} denote test and training set perplexity; D is the number of words in the training data; $\tilde{\lambda}_i$ are *regularized* (*i.e.*, smoothed) estimates of the model parameters; and γ is a constant independent of domain, training set size, and model type [2], [3]. Thus, one can improve test performance by shrinking the parameter sum $\sum_i |\tilde{\lambda}_i|$, and both Model M and rMDI models are designed to improve upon word n-gram models in this way. However, as training set size increases, the last term in eq. (1) tends to grow smaller, which suggests the gain to be had by shrinking parameter values will also decrease. Thus, it is uncertain whether Model M and rMDI models will retain their performance improvements over word n-gram models on larger training corpora.

The outline of this paper is as follows: In Section II, we review Model M and rMDI models. In Section III, we elaborate on why performance gains decrease as training sets grow, and show how gains vary for some actual models. In Section IV, we examine the task of model combination when using Model M and rMDI models, as this is a key issue when tackling large-scale domains. In Section V, we apply these methods to a variety of medium-to-large-scale tasks. For more details about this work, see [4].

II. Background

In this section, we review Model M and rMDI models as well as the results for performance prediction for exponential language models given in [2], [3]. An exponential model $p_\Lambda(y|x)$ is a model with a set of features $\{f_i(x,y)\}$ and equal number of parameters $\Lambda = \{\lambda_i\}$ where

$$p_\Lambda(y|x) = \frac{\exp(\sum_i \lambda_i f_i(x,y))}{\sum_{y'} \exp(\sum_i \lambda_i f_i(x,y'))} \tag{2}$$

Remarkably, eq. (1) holds for many exponential language models including Model M and rMDI models; the relationship is strongest if the $\tilde{\Lambda} = \{\tilde{\lambda}_i\}$ are estimated using $\ell_1 + \ell_2^2$ regularization [5]; *i.e.*, parameters are chosen to optimize

$$\mathcal{O}_{\ell_1+\ell_2^2}(\Lambda) = \log \text{PP}_{\text{train}} + \frac{\alpha}{D} \sum_i |\lambda_i| + \frac{1}{2\sigma^2 D} \sum_i \lambda_i^2 \tag{3}$$

for some α and σ. When using natural logs in eq. (1) and taking $(\alpha = 0.5, \sigma^2 = 6)$, the constant $\gamma = 0.938$ yields a mean error equivalent to a few percent in perplexity over the models evaluated in [3]. These values of α and σ also yield good test set performance over a wide variety of training sets.

It follows that if one can *shrink* the "size" of a model (proportional to $\sum_i |\lambda_i|$) while not damaging training set

performance, test set performance should improve. In [1], we use this reasoning to motivate Model M and rMDI models. Model M is a class-based n-gram model that can be viewed as the result of shrinking an exponential word n-gram model using word classes. If we assume each word w is mapped to a single class $c(w)$, we can write

$$p(w_1 \cdots w_l) = \prod_{j=1}^{l+1} p(c_j|c_1 \cdots c_{j-1}, w_1 \cdots w_{j-1}) \times \prod_{j=1}^{l} p(w_j|c_1 \cdots c_j, w_1 \cdots w_{j-1}) \quad (4)$$

where c_{l+1} is the end-of-sentence token. Let f_θ denote a binary n-gram feature such that $f_\theta(x, y) = 1$ iff xy "ends" in the n-gram θ. Let $p_{\text{ng}}(y|\theta)$ denote an exponential n-gram model, where we have a feature $f_{\theta'}$ for each suffix θ' of each θy occurring in the training set. For example, the model $p_{\text{ng}}(w_j|w_{j-1}c_j)$ has a feature f_θ for each n-gram θ in the training set of the form w_j, $c_j w_j$, or $w_{j-1}c_j w_j$. Let $p_{\text{ng}}(y|\theta_1, \theta_2)$ denote a model containing all features in $p_{\text{ng}}(y|\theta_1)$ and $p_{\text{ng}}(y|\theta_2)$. Then, we can define (the trigram version of) Model M as

$$\begin{aligned} p(c_j|c_1 \cdots c_{j-1}, w_1 \cdots w_{j-1}) &\equiv p_{\text{ng}}(c_j|c_{j-2}c_{j-1}, w_{j-2}w_{j-1}) \\ p(w_j|c_1 \cdots c_j, w_1 \cdots w_{j-1}) &\equiv p_{\text{ng}}(w_j|w_{j-2}w_{j-1}c_j) \end{aligned} \quad (5)$$

Regularized minimum discrimination information (rMDI) models can be viewed as the result of shrinking an exponential model using a prior distribution. Minimum discrimination information (MDI) models [6] have the form

$$p_\Lambda(y|x) = \frac{q(y|x) \exp(\sum_i \lambda_i f_i(x, y))}{\sum_{y'} q(y'|x) \exp(\sum_i \lambda_i f_i(x, y'))} \quad (6)$$

for some prior distribution $q(y|x)$. While regularization is not used in [6], we found in [2] that when regularizing $p_\Lambda(y|x)$ in the way described earlier, eq. (1) holds for these models if $q(y|x)$ is ignored in computing model size (assuming $q(y|x)$ is estimated on an independent training corpus). Regularized MDI models are well-suited to the task of domain adaptation, where one has a test set and small training set from one domain, and a large training set from a different domain. One can build a language model on the outside domain, and use this model as the prior when building a model on the in-domain data. While exponential models of any form can be used in eq. (6), [1] evaluated the use of exponential word n-gram models, *i.e.*, models of the form $p_{\text{ng}}(w_j|w_{j-2}w_{j-1})$ (for trigrams). In this paper, we also evaluate a variant of rMDI, *cascaded rMDI*, that can be used to combine an arbitrary number of training corpora rather than just two. In this method, one orders the available corpora from most "out-of-domain" to most "in-domain" and applies the rMDI technique repeatedly.

III. Analyzing How Models Scale

In this section, we discuss why it's important to study how models scale with training set size, *i.e.*, why good performance on small data sets often does not carry over to large ones. One obvious reason to worry about this issue is that many algorithms in the literature have been shown not to scale well. Here, we show how to explain this phenomenon for many types of models by using eq. (1), and study how this effect affects Model M and rMDI models by plotting relative performances over a variety of training set sizes.

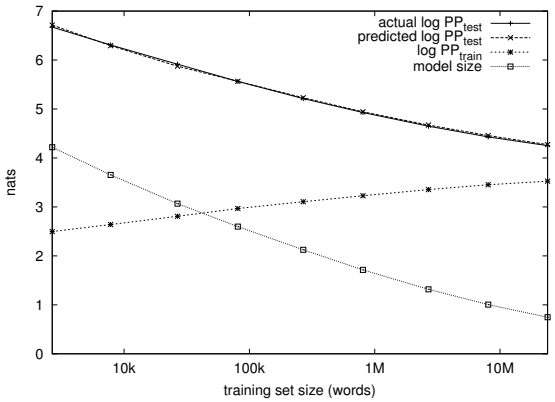

Fig. 1. Predicted and actual $\log \text{PP}_{\text{test}}$, $\log \text{PP}_{\text{train}}$, and model size ($\frac{\gamma}{D} \sum_i |\tilde{\lambda}_i|$) for word trigram models built on varying amounts of WSJ data.

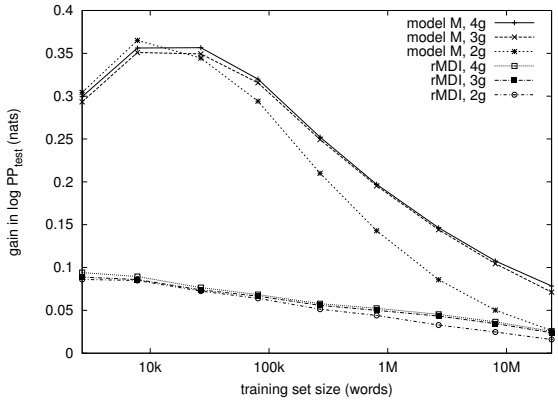

Fig. 2. Gains in $\log \text{PP}_{\text{test}}$ for Model M and rMDI models as compared to a word n-gram baseline, for bigram, trigram, and 4-gram models built on varying amounts of WSJ data. For rMDI, the out-of-domain corpus is Broadcast News text and is the same length as the in-domain WSJ corpus.

If we define the *size* of a model p_Λ to be $\frac{\gamma}{D} \sum_i |\tilde{\lambda}_i|$, eq. (1) tells us that test performance (in log PP) is approximately equal to the sum of training performance (in log PP) and model size. In Figure 1, we graph these quantities over varying amounts of training data (from 100 to 900k sentences) for an exponential word trigram model built on Wall Street Journal (WSJ) data with a 21k word vocabulary. While the sum $\sum_i |\tilde{\lambda}_i|$ grows with more data, it grows slower than D, so the overall model size tends to decrease as we go to the right.

This is significant because Model M and rMDI models achieve their performance improvements over n-gram models by shrinking model size, and it seems likely that if n-gram model sizes decrease, so will the shrinkage gain. In the limit of infinite data, we expect the size of a trigram model, say, to go to zero and hence expect no improvement from the corresponding Model M or rMDI models. These models condition their predictions on exactly two words of history, so

TABLE I
WORD-ERROR RATES FOR VARIOUS METHODS FOR DOMAIN ADAPTATION.

	in-domain (WSJ) training set (sents.)			
	1k	10k	100k	900k
word n-gram models				
WSJ only				
KN n-gram	34.5%	30.4%	26.1%	22.6%
exp. n-gram	34.6%	30.3%	25.7%	22.5%
WSJ and BN, 1:1 ratio				
interp	34.3%	30.0%	25.4%	22.3%
merge	34.1%	29.6%	25.0%	22.1%
rMDI	34.0%	29.6%	25.1%	22.1%
WSJ and BN and SWB, 1:3:10 ratio				
interp	33.8%	29.7%	25.0%	
merge	33.3%	29.3%	25.2%	
casc. rMDI	33.1%	28.7%	24.6%	
Model M				
WSJ only				
Model M	35.3%	29.1%	24.2%	21.5%
WSJ and BN, 1:1 ratio				
interp	33.9%	28.3%	23.9%	21.2%
merge	33.9%	28.2%	23.9%	21.2%
rMDI	34.8%	28.6%	23.8%	21.2%

TABLE II
WORD-ERROR RATES FOR VARIOUS METHODS FOR MODEL COMBINATION.

word n-gram models			*Model M*		
	PP	WER		PP	WER
interp+, rMDI	180.8	14.3%	interp+, rMDI	169.3	13.7%
interp, rMDI	181.2	14.4%	interp+	169.4	13.6%
interp+, exp.	184.9	14.4%	interp, rMDI	170.1	13.7%
interp, KN	190.6	14.5%	merge	175.0	13.8%
merge, KN	194.2	14.6%	interp	175.3	13.7%
casc. rMDI	210.1	14.8%	casc. rMDI	203.3	14.2%

they can do no better than an "ideal" trigram model. Hence, the issue is not *whether* Model M and rMDI model gains will disappear as training set size grows, but *when*.

In Figure 2, we display gain in $\log \text{PP}_{\text{test}}$ relative to a word n-gram model for Model M and rMDI models for $n \in \{2, 3, 4\}$. As predicted, gains generally decrease as the training set expands. Similarly, gains are smaller for smaller n since n-gram model sizes shrink as n shrinks.

At the right edge of the graph, the gains for most algorithms are 0.03 nats or less, where a nat is a "natural" bit, or $\log_2 e$ regular bits. Each 0.01 nat difference corresponds to about 1% in PP. However, the gain for 4-gram Model M is 0.08 nats, which translates to a 1.4% absolute reduction in word-error rate (22.2% $\Rightarrow$ 20.8%) using the ASR setup described in Section IV-A. While gain is dropping as training set size increases, Model M still appears promising for data sets substantially larger than 900k sentences.

IV. SCALING MODEL COMBINATION

For large-scale domains, one typically has language model training data from many different sources; for example, the IBM GALE Arabic ASR system uses 16 separate corpora. Furthermore, these corpora generally differ in relevance and amount, and aggregating the data into a single corpus may not work best. Thus, a central issue in handling large domains is how to best combine multiple data sets or models. In this section, we attempt to discover the best methods for combining Model M models and to characterize when rMDI modeling can improve model combination performance. We use small and medium-sized data sets so we can evaluate a large number of methods under a large number of conditions, and attempt to predict performance on large tasks via extrapolation. We use these findings to inform which algorithms to assess in the large-scale experiments in Section V.

The best way to combine data or models will depend on the relationship between the training and test corpora, so we investigate two different scenarios. In Section IV-A, we consider a typical domain adaptation task where we have a modest amount of training data from the same domain as the test data, and equal or larger amounts of out-of-domain data. In Section IV-B, we consider a model combination task where we have many corpora from similar domains as the test data.

A. Domain Adaptation

These ASR experiments are an expanded version of the domain adaptation experiments in [1]; here, we consider more corpora, larger data sets, and more algorithms. The acoustic model is built from 50h of Broadcast News data and contains 2176 context-dependent states and 50k Gaussians. We evaluate language models via lattice rescoring of lattices generated using a small trigram language model. We use a 47k-word WSJ test set and in-domain WSJ training sets of various sizes. For the out-of-domain data, we consider the cases where only Broadcast News (BN) data is available and where both BN and Switchboard (SWB) data are available.

We compare the techniques of *linear interpolation*, *count merging*, and rMDI modeling. In linear interpolation, separate language models are built on each corpus and linearly interpolated, with interpolation weights being optimized on a held-out set. In count merging, the component corpora are concatenated into a single corpus, and a single language model is built on the merged data set. Unlike in linear interpolation where each model is assigned a fixed weight independent of history for each word prediction, count merging can be viewed as assigning a weight proportional to the history count of each model. In contrast, rMDI modeling can be viewed as backing off from the in-domain model to the out-of-domain model.

In Table I, we display a subset of our ASR WER results; complete results can be found in [4]. The top part of the table corresponds to word n-gram models, while the bottom part corresponds to Model M. Each column represents a different in-domain training set size. Each subsection of the table corresponds to using a different amount of out-of-domain data. For example, the *WSJ and BN and SWB, 1:3:10 ratio* section corresponds to using a BN corpus three times larger than the in-domain data and a SWB corpus ten times larger than the in-domain data. All of the word n-gram models are exponential n-gram models except for the first row, which corresponds to a conventional word n-gram model with modified Kneser-Ney smoothing [7]. We use the trigram versions of each model.

Unlike in Section III, we induce word classes on the given training set(s), rather than always using word classes from the largest training set. We note that it is straightforward to combine rMDI domain adaptation with Model M; one can

TABLE III
COMPARISON OF LANGUAGE MODELS ON THE VOICEMAIL TRANSCRIPTION TASK.

word n-gram models

	WER
interp, KN n-gram	16.9%
rMDI, exp. n-gram	16.7%
merge, exp. n-gram	16.6%
interp, exp. n-gram	16.6%

Model M

	WER
rMDI	16.4%
merge	16.3%
interp	16.3%

TABLE IV
WORD-ERROR RATES FOR INTERPOLATED LMS ON SEVERAL GALE ARABIC TEST SETS, VARYING HOW MANY COMPONENT MODELS ARE WORD n-GRAM MODELS AND HOW MANY ARE MODEL M.

	DEV07	DEV08	EVAL08
Interpolation over all 16 corpora			
Baseline: 16 KN LMs	9.5%	11.0%	9.4%
5 Model M + 11 KN LMs	9.1%	10.6%	9.0%
3 M (500c) + 2 M (150c) + 11 KN	9.0%	10.4%	8.9%
Interpolation over 5 of 16 corpora			
5 KN LMs	10.0%	11.3%	9.6%
5 Model M LMs	9.4%	10.8%	9.0%

simply do rMDI domain adaptation separately for each of the two component models given in eq. (5), as long as the same word classes are used everywhere.

For word n-gram models, the rMDI methods generally perform best or near best in all conditions. While WER gains for rMDI over interpolation can be as large as 1% absolute, the difference between techniques when using 900k sentences of in-domain data is much smaller. Intuitively, the backoff-like behavior of rMDI should be well-suited to domain adaptation, as it seems reasonable that in-domain counts should take priority over out-of-domain counts, when present.

Overall, Model M outperforms word n-gram models for all of the training sets except the smallest, and gains from domain adaptation are comparable to those for word n-gram models. However, with Model M, rMDI does not perform particularly well, and no one algorithm dominates the others. For the 900k-sentence in-domain training set, there is no significant difference between algorithms. In summary, for larger training sets, we hypothesize that when combining word n-gram models for domain adaptation, rMDI may yield small gains over other methods; for Model M, we predict that all methods will perform about equally.

B. Model Combination

In these experiments, we use the same data sets as in the English Broadcast News task described in Section V-B, except we subsample each training set to $\frac{1}{10}$th its size and build trigram versions of each model instead of 4-gram models. There are a total of six training corpora ranging in size from 170k words to 14.7M words after subsampling; each contains Broadcast News data of some sort. Thus, this task is qualitatively different from our domain adaptation task, where some corpora are clearly in-domain and others are not. We evaluate the same algorithms as in the last section, as well as a few more. To order the training corpora for cascaded rMDI, we build n-gram models on each corpus and compute the perplexity of an in-domain held-out set to guide us.

One unappealing aspect of linear interpolation is that when one of the component models has no counts for a particular history (while the others do), it still gets its full prediction weight. We can attempt to improve prediction in this situation by combining each component model with a "general" model built on all of the training data combined, *i.e.*, the count-merged model. We consider two different ways of combining each corpus-specific model with the general model: linear interpolation and rMDI modeling. In linear interpolation, interpolating each component model with the general model is equivalent to just adding the general model into the overall interpolation. In rMDI modeling, we use the general model as the prior when training each corpus-specific model.

In Table II, we display development set PP and test set WER for a variety of model combination algorithms applied to both word n-gram models and Model M. The notation *interp+* refers to doing interpolation where the general/count-merged model is included in the mix; *exp.* means exponential n-gram models whereas *KN* refers to conventional n-gram models; and *rMDI* (with interpolation) refers to training each corpus-specific model using the general model as a prior.

The most popular model combination techniques are linear interpolation and count merging with conventional n-gram models, yielding a WER of 14.5% and 14.6%, respectively. The algorithm yielding the best performance on the development set is *interp+, rMDI*, giving a WER of 14.3% for word n-gram models and 13.7% for Model M. However, a WER of 13.7% can also be achieved through simple linear interpolation with Model M. In summary, we speculate that for large training sets when using word n-gram models, small gains over simple interpolation may be possible with *interp+, rMDI*. With Model M, simple linear interpolation is the easiest to implement and performs as well as any other method.

V. EXPERIMENTS

In this section, we investigate whether Model M and rMDI modeling can improve the performance of existing medium and large-scale state-of-the-art systems. For each system, we compare against the current best language model for that system trained on all available training data; except where noted, this is the system we refer to as the baseline. We evaluate the best methods found in Section IV-B, but also do contrast runs with other methods to attempt to confirm the findings in that section. While Model M gives consistent gains over word n-gram models in Section IV-B, we verify whether these gains carry over to larger data sets.

All exponential models are trained with $\ell_1 + \ell_2^2$ regularization with $(\alpha = 0.5, \sigma^2 = 6)$; conventional n-gram models are trained using modified Kneser-Ney (KN) smoothing [7]. Unless otherwise noted, we use the 4-gram version of each model; we induce 150 word classes using the algorithm of [8] for Model M; and interpolation weights are trained to optimize the perplexity of a held-out set. Experiments with Model M are substantially more expensive in both time and memory than those with n-gram models, partially due to algorithmic considerations and partially because our exponential model

TABLE V
BLEU SCORES FOR VARIOUS LANGUAGE MODELS FOR IRAQI ARABIC/ENGLISH AND SPANISH/ENGLISH TRANSLATION, USING N-BEST LIST RESCORING OF N-BEST LISTS OF VARIOUS SIZE. FOR EACH MODEL, WE REPORT (DEVELOPMENT SET/TEST SET) RESULTS.

	50-best	20-best	10-best
English ⇒ Iraqi Arabic			
3-gram	30.6/29.7	30.6/29.7	30.6/29.7
Model M	31.0/30.3	31.1/30.4	31.1/30.2
3-gram + Model M	31.0/30.5	31.0/30.3	31.0/30.2
Iraqi Arabic ⇒ English			
3-gram	25.4/24.7	25.4/24.7	25.4/24.7
Model M	24.9/26.0	25.1/25.8	25.3/25.9
3-gram + Model M	25.5/26.1	25.4/26.0	25.5/26.0
English ⇒ Spanish			
4-gram	21.7/21.5	21.7/21.5	21.7/21.2
Model M	22.8/23.0	22.7/23.0	22.8/22.8
4-gram + Model M	22.6/22.9	22.5/22.8	22.8/22.8
Spanish ⇒ English			
4-gram	18.1/17.6	18.3/17.6	18.0/17.6
Model M	19.6/18.4	19.3/18.8	19.1/18.3
4-gram + Model M	19.4/18.5	19.2/18.8	19.0/18.4

code has not yet been optimized much. This constrained the number of Model M experiments we were able to run.

A. English Voicemail Transcription

We evaluate the performance of Model M and rMDI models on the task of English voicemail transcription. Recently, ASR is increasingly being deployed in unified messaging systems to serve as an aid to human transcribers or as a standalone service. Here, we report on an in-house voicemail transcription task. The ASR system is based on the 2007 IBM GALE speech transcription system [9]. The discriminatively-trained acoustic model was trained on 2000h of voicemail messages and contains 8000 context-dependent states and 300k Gaussians.

We have two sources of language model data: the verbatim transcripts of the acoustic training data (17M words), and 41M words of approximate voicemail transcripts cleaned up for readability. The first corpus is very well-matched to the test set; the second corpus less so. The baseline language model, built using a 40k-word lexicon, is the interpolation of two word 4-gram models, one trained on each of the LM training corpora. The 5.5h test set consists of 900 messages and 62k words; the perplexity of the baseline LM on this set is 43 and the WER is 16.9%. Language models are evaluated via lattice rescoring on lattices generated using the baseline LM.

To decide which model combination method should work best with Model M, the main issue is whether the two corpora are similar enough to be considered a single corpus or not. If so, we expect count merging to do best; if not, we expect linear interpolation to do as well as anything else. In Table III, we display the results for various algorithms. Model M yields the best performance; a WER of 16.3% is obtained both through count merging and interpolation (using the same weights as the baseline model), a gain of 0.6% absolute.

B. English Broadcast News Transcription

In this section, we examine whether Model M can improve performance on an English Broadcast News task. The ASR system is based on the 2007 IBM GALE speech transcription system [9]. The discriminatively-trained acoustic model was trained on 430h of Broadcast News audio and contains 6000 context-dependent states and 250k Gaussians. The LM training text consists of a total of 400M words from the following six sources: 1996 CSR Hub4 language model data; EARS BN03 closed captions; GALE Phase 2 Distillation GNG Evaluation Supplemental Multilingual data; Hub4 acoustic model training transcripts; TDT4 closed captions; and TDT4 newswire. The vocabulary is 80k words and the baseline language model is a linear interpolation of word 4-gram models, one for each corpus. Interpolation weights are chosen to optimize perplexity on a held-out set of 25k words, the rt03 evaluation set. The evaluation set is the 2.5h rt04 evaluation set containing 45k words; the WER of the baseline LM on this data set is 13.0%.

The experiments in Section IV-B use a scaled-down version of this task, and thus we expect the same methods will work best. We build Model M on each source and interpolate them using the same weights as in the baseline, yielding a WER of 12.3%, or a gain of 0.7% absolute. As far as we know, this is the best single-system result for this data set, surpassing the previous best of 12.6% [10]. On the held-out set, the perplexity is reduced from 133 for the baseline to 121. As a contrast, we also evaluated cascaded rMDI for model combination, ordering models by their interpolation weight. This model performed much worse as in Section IV-B, yielding a WER of 13.1% and perplexity of 150.

C. GALE Arabic Transcription

Arabic broadcast transcription is a core component of DARPA's Global Autonomous Language Exploitation (GALE) program. In this section, we assess whether Model M can improve the performance of the best Arabic ASR system fielded in the January 2009 GALE evaluation. The acoustic model is a discriminatively-trained Universal Background Model [11] trained on 1400h of transcribed audio [12]. We have 16 sources of language model training data totaling 1.3 billion words: transcripts of the audio data; the Arabic Gigaword corpus; newsgroup and weblog data; etc. The baseline language model has a vocabulary of 774k words and is a linear interpolation of 4-gram models built on each of the 16 sources.

In our initial experiment, we build Model M models on the five corpora with the highest interpolation weights in the baseline model, with a combined weight of 0.6. We replace the corresponding n-gram models with Model M for these five sources and reoptimize interpolation weights. In the first two rows of Table IV, we present lattice rescoring results for the baseline LM and this new LM over a variety of test sets: DEV07 (2.6h), DEV08 (3h) and EVAL08 (3h). We see that a significant improvement of 0.4% absolute is achieved. To isolate the gains of Model M, we also display results when interpolating only the five sources under consideration. In the last two rows of Table IV, we show results for interpolating only conventional n-gram models and only Model M models; we see 0.5–0.6% absolute gain from Model M.

Given that our Arabic vocabulary is much larger than the original WSJ vocabulary used to optimize the number of word classes, we investigate whether using more than 150 word classes can improve performance. On the corpus with the highest interpolation weight in the baseline LM (Broadcast News audio transcripts, 5M words), we vary the number of word classes used with Model M and find that 500 word classes yield the best results. We rebuild three of the five Model M models in the 16-way interpolation from before using 500 classes instead of 150, and this yields additional improvement as seen from the third row in Table IV. For reference, our best previous LM included interpolation with a 6-gram neural net LM and yielded WER's of 9.3%, 10.6%, and 9.1% on our three test sets.

D. Machine Translation

In this section, we evaluate whether Model M performs well on the task of machine translation. In addition, we evaluate whether the performance of Model M can be improved by linearly interpolating with a word n-gram model. We consider two different domains, Iraqi Arabic/English and Spanish/English bidirectional translation. For Iraqi Arabic/English, the parallel training corpus consists of 430k utterance pairs containing 98k unique Arabic words and 31k unique English words. The Arabic LM training data is composed of the 2.7M words of Arabic in the parallel training corpus. For English, we use 6.4M words of text, of which the English data in the MT training corpus is a subset. For English to Arabic, we have a development set of 19k words to tune feature weights, and a test set of about the same size. For Arabic to English, the development and test sets are about 21k words.

For Spanish/English, the target task is a travel application. The MT training data consists of conversational travel data as well as movie subtitles and TV show transcriptions, 2.1M sentence pairs in all with 14.3M English tokens (137k unique) and 13.5M Spanish tokens (176k unique). The MT training data is also used for language model training. The test and development sets consist of 711 sentence pairs each, with about 5.9k English and 5.6k Spanish tokens in each. We use a phrase-based multi-stack decoder using log-linear models similar to Pharaoh [13]. We include features for bidirectional translation probabilities, bidirectional lexicon weights, language model scores, distortion model scores, and sentence length penalty.

To evaluate Model M, we do N-best list rescoring and measure translation performance using BLEU score with one reference for each hypothesis. The baseline language model is a conventional n-gram model, and this baseline model is used to generate translation N-best lists of various size (N=10, 20, and 50). Feature weights (including the language model weight) are optimized on the development data using the downhill simplex method to maximize BLEU score. In addition to the baseline, we evaluate Model M as well as Model M interpolated with the baseline n-gram model. For Arabic/English, the trigram versions of each model are used due to the small amount of training data, over morphemes for Iraqi Arabic and over words for English.

In Table V, we display the BLEU scores for each model for each different N-best list size, for both the development and test sets. We see consistent gains in test set BLEU scores across all conditions for Model M as compared to the baseline, with gains ranging from 0.5 to 1.6 points. Interpolating Model M with the baseline gives about the same performance as Model M alone, indicating that Model M already encompasses most or all of the information included in an n-gram model.

VI. Discussion

We show that Model M consistently outperforms the best existing language models over a variety of domains and applications. While our analysis shows that shrinkage-based gains will decrease as training sets increase in size, we still find significant gains even on tasks where over a billion words of training data are available. We achieve WER gains of 0.5–0.7% absolute for three large-scale ASR systems, including state-of-the-art systems on the highly competitive English Broadcast News and GALE Arabic tasks. On the other hand, while rMDI models can give gains against other techniques for domain adaptation on moderately-sized corpora, it does not outperform simple linear interpolation on large data sets. In summary, despite the advances in language modeling over the past decades, word n-gram models remain the technology of choice in systems both large and small. Here, we show that Model M is a compelling alternative for a wide range of applications and operating points.

References

[1] S. F. Chen, "Shrinking exponential language models," in *Proc. of NAACL-HLT*, 2009.

[2] ——, "Performance prediction for exponential language models," in *Proc. NAACL-HLT*, 2009.

[3] ——, "Performance prediction for exponential language models," IBM Research Division, Tech. Rep. RC 24671, October 2008.

[4] S. F. Chen, L. Mangu, B. Ramabhadran, R. Sarikaya, and A. Sethy, "Scaling shrinkage-based language models," IBM Research Division, Tech. Rep. In preparation, July 2009.

[5] J. Kazama and J. Tsujii, "Evaluation and extension of maximum entropy models with inequality constraints," in *Proc. EMNLP*, 2003.

[6] S. Della Pietra, V. Della Pietra, R. L. Mercer, and S. Roukos, "Adaptive language modeling using minimum discriminant estimation," in *Proc. the Speech and Natural Language DARPA Workshop*, February 1992.

[7] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," Harvard U., Tech. Rep. TR-10-98, 1998.

[8] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, December 1992.

[9] S. F. Chen, B. Kingsbury, L. Mangu, D. Povey, G. Saon, H. Soltau, and G. Zweig, "Advances in speech transcription at IBM under the DARPA EARS program," *IEEE Trans. on Audio, Speech and Language Processing*, pp. 1596–1608, 2006.

[10] M. J. F. Gales, D. Y. Kim, P. C. Woodland, H. Y. Chan, D. Mrva, R. Sinha, and S. E. Tranter, "Progress in the CU-HTK broadcast news transcription system," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1513–1525, September 2006.

[11] D. Povey, S. M. Chu, and B. Varadarajan, "Universal background model based speech recognition," in *Proc. ICASSP*, 2008.

[12] H. Soltau, G. Saon, B. Kingsbury, J. Kuo, L. Mangu, D. Povey, and G. Zweig, "The IBM 2006 GALE Arabic ASR system," in *Proc. of ICASSP*, 2007.

[13] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proc. HLT-NAACL*. Morristown, NJ, USA: Association for Computational Linguistics, 2003, pp. 48–54.