N-gram Weighting: Reducing Training Data Mismatch in Cross-Domain Language Model Estimation

Bo-June (Paul) Hsu, James Glass
MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street, Cambridge, MA, 02139 USA
{bohsu,glass}@csail.mit.edu

Abstract

In domains with insufficient matched training data, language models are often constructed by interpolating component models trained from partially matched corpora. Since the n-grams from such corpora may not be of equal relevance to the target domain, we propose an n-gram weighting technique to adjust the component n-gram probabilities based on features derived from readily available segmentation and metadata information for each corpus. Using a log-linear combination of such features, the resulting model achieves up to a 1.2% absolute word error rate reduction over a linearly interpolated baseline language model on a lecture transcription task.

1 Introduction

Many application domains in machine learning suffer from a dearth of matched training data. However, partially matched data sets are often available in abundance. Past attempts to utilize the mismatched data for training often result in models that exhibit biases not observed in the target domain. In this work, we will investigate the use of the often readily available data segmentation and metadata attributes associated with each corpus to reduce the effect of such bias. We will examine this approach in the context of language modeling for lecture transcription.

Compared with other types of audio data, lecture speech often exhibits a high degree of spontaneity and focuses on narrow topics with special terminologies (Glass et al., 2004). While we may have existing transcripts from general lectures or written text on the precise topic, training data that matches both the style and topic of the target lecture is often scarce. Thus, past research has investigated various adaptation and interpolation techniques to make use of partially matched corpora (Bellegarda, 2004).

Training corpora are often segmented into documents with associated metadata, such as title, date, and speaker. For lectures, if the data contains even a few lectures on linear algebra, conventional language modeling methods that lump the documents together will tend to assign disproportionately high probability to frequent terms like *vector* and *matrix*. Can we utilize the segmentation and metadata information to reduce the biases resulting from training data mismatch?

In this work, we present such a technique where we weight each n-gram count in a standard n-gram language model (LM) estimation procedure by a relevance factor computed via a log-linear combination of n-gram features. Utilizing features that correlate with the specificity of n-grams to subsets of the training documents, we effectively de-emphasize out-of-domain n-grams. By interpolating models, such as general lectures and course textbook, that match the target domain in complementary ways, and optimizing the weighting and interpolation parameters jointly, we allow each n-gram probability to be modeled by the most relevant interpolation component. Using a combination of features derived from multiple partitions of the training documents, the resulting weighted n-gram model achieves up to a 1.2% absolute word error rate (WER) reduction over a linearly interpolated baseline on a lecture transcription task.

Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 828–837,
Honolulu, October 2008. ©2008 Association for Computational Linguistics

2 Related Work

To reduce topic mismatch in LM estimation, we (2006) have previously assigned topic labels to each word by applying HMM-LDA (Griffiths et al., 2005) to the training documents. Using an ad hoc method to reduce the effective counts of n-grams ending on topic words, we achieved better perplexity and WER than standard trigram LMs. Intuitively, de-emphasizing such n-grams will lower the transition probability to out-of-domain topic words from the training data. In this work, we further explore this intuition with a principled feature-based model, integrated with LM smoothing and estimation to allow simultaneous optimization of all model parameters.

As Gao and Lee (2000) observed, even purported matched training data may exhibit topic, style, or temporal biases not present in the test set. To address the mismatch, they partition the training documents by their metadata attributes and compute a measure of the likelihood that an n-gram will appear in a new partitioned segment. By pruning n-grams with generality probability below a given threshold, the resulting model achieves lower perplexity than a count-cutoff model of equal size. Complementary to our work, this technique also utilizes segmentation and metadata information. However, our model enables the simultaneous use of all metadata attributes by combining features derived from different partitions of the training documents.

3 N-gram Weighting

Given a limited amount of training data, an n-gram appearing frequently in a single document may be assigned a disproportionately high probability. For example, an LM trained from lecture transcripts tends to assign excessive probability to words from observed lecture topics due to insufficient coverage of the underlying document topics. On the other hand, excessive probabilities may also be assigned to n-grams appearing consistently across documents with mismatched style, such as the course textbook in the written style. Traditional n-gram smoothing techniques do not address such issues of insufficient topic coverage and style mismatch.

One approach to addressing the above issues is to weight the counts of the n-grams according to the concentration of their document distributions. Assigning higher weights to n-grams with evenly spread distributions captures the style of a data set, as reflected across all documents. On the other hand, emphasizing the n-grams concentrated within a few documents focuses the model on the topics of the individual documents.

In theory, n-gram weighting can be applied to any smoothing algorithm based on counts. However, because many of these algorithms assume integer counts, we will apply the weighting factors to the smoothed counts, instead. For modified Kneser-Ney smoothing (Chen and Goodman, 1998), applying n-gram weighting yields:

$$p(w|h) = \frac{\beta(hw)\tilde{c}'(hw)}{\sum_w \beta(hw)\tilde{c}(hw)} + \alpha(h)p(w|h')$$

where $p(w|h)$ is the probability of word w given history h, $\tilde{c}$ is the adjusted Kneser-Ney count, $\tilde{c}'$ is the discounted count, β is the n-gram weighting factor, α is the normalizing backoff weight, and h' is the backoff history.

Although the weighting factor β can in general be any function of the n-gram, in this work, we will consider a log-linear combination of n-gram features, or $\beta(hw) = \exp(\mathbf{\Phi}(hw) \cdot \boldsymbol{\theta})$, where $\mathbf{\Phi}(hw)$ is the feature vector for n-gram hw and $\boldsymbol{\theta}$ specifies the parameter vector to be learned. To better fit the data, we allow independent parameter vectors $\boldsymbol{\theta}_o$ for each n-gram order o. Note that with $\beta(hw) = 1$, the model degenerates to the original modified Kneser-Ney formulation. Furthermore, β only specifies the relative weighting among n-grams with a common history h. Thus, scaling $\beta(hw)$ by an arbitrary function $g(h)$ has no effect on the model.

In isolation, n-gram weighting shifts probability mass from out-of-domain n-grams via backoff to the uniform distribution to improve the generality of the resulting model. However, in combination with LM interpolation, it can also distribute probabilities to LM components that better model specific n-grams. For example, n-gram weighting can de-emphasize off-topic and off-style n-grams from general lectures and course textbook, respectively. Tuning the weighting and interpolation parameters jointly further allows the estimation of the n-gram probabilities to utilize the best matching LM components.

3.1 Features

To address the issue of sparsity in the document topic distribution, we can apply n-gram weighting with features that measure the concentration of the n-gram distribution across documents. Similar features can also be computed from documents partitioned by their categorical metadata attributes, such as *course* and *speaker* for lecture transcripts. Whereas the features derived from the corpus documents should correlate with the topic specificity of the n-grams, the same features computed from the speaker partitions might correspond to the speaker specificity. By combining features from multiple partitions of the training data to compute the weighting factors, n-gram weighting allows the resulting model to better generalize across categories.

To guide the presentation of the n-gram features below, we will consider the following example partition of the training corpus. Words tagged by HMM-LDA as topic words appear in bold.

A B A	A **C** **C** A	**B** A B
B A **A** C	C B A	A B A
A C B	A A **C**	A **B** **B** A

One way to estimate the specificity of an n-gram across partitions is to measure the n-gram frequency f, or the fraction of partitions containing an n-gram. For instance, $f(\mathrm{A}) = 3/3$, $f(\mathrm{C}) = 2/3$. However, as the size of each partition increases, this ratio increases to 1, since most n-grams have a non-zero probability of appearing in each partition. Thus, an alternative is to compute the normalized entropy of the n-gram distribution across the S partitions, or $h = \frac{-1}{\log S} \sum_{s=1}^{S} p(s) \log p(s)$, where $p(s)$ is the fraction of an n-gram appearing in partition s. For example, the normalized entropy of the unigram C is $h(\mathrm{C}) = \frac{-1}{\log 3}[\frac{2}{6} \log \frac{2}{6} + \frac{4}{6} \log \frac{4}{6} + 0] = .58$. N-grams clustered in fewer partitions have lower entropy than ones that are more evenly spread out.

Following (Hsu and Glass, 2006), we also consider features derived from the HMM-LDA word topic labels.[1] Specifically, we compute the empirical probability t that the target word of the n-gram is labeled as a topic word. In the example corpus, $t(\mathrm{C}) = 3/6$, $t(\mathrm{A\ C}) = 2/4$.

Feature	of the	i think	k means	the sun	this is a	a lot of	big o of	e m f
Random	0.03	0.32	0.33	0.19	0.53	0.24	0.37	0.80
$\log(c)$	9.29	8.09	3.47	5.86	6.82	7.16	3.09	4.92
f^{doc}	1.00	0.93	0.00	0.18	0.92	0.76	0.00	0.04
f^{course}	1.00	1.00	0.06	0.56	0.94	0.94	0.06	0.06
f^{speaker}	0.83	0.70	0.00	0.06	0.41	0.55	0.01	0.00
h^{doc}	0.96	0.84	0.00	0.56	0.93	0.85	0.00	0.34
h^{course}	0.75	0.61	0.00	0.55	0.78	0.65	0.00	0.00
h^{speaker}	0.76	0.81	0.00	0.09	0.65	0.80	0.12	0.00
t^{doc}	0.00	0.00	0.91	1.00	0.01	0.00	0.00	0.04
t^{course}	0.00	0.00	0.88	0.28	0.01	0.00	0.00	1.00
t^{speaker}	0.00	0.00	0.94	0.92	0.01	0.00	0.09	0.99

Table 1: A list of n-gram weighting features. f: n-gram frequency, h: normalized entropy, t: topic probability.

All of the above features can be computed for any partitioning of the training data. To better illustrate the differences, we compute the features on a set of lecture transcripts (see Section 4.1) partitioned by lecture (doc), course, and speaker. Furthermore, we include the log of the n-gram counts c and random values between 0 and 1 as baseline features. Table 1 lists all the features examined in this work and their values on a select subset of n-grams.

3.2 Training

To tune the n-gram weighting parameters $\boldsymbol{\theta}$, we apply Powell's method (Press et al., 2007) to numerically minimize the development set perplexity (Hsu and Glass, 2008). Although there is no guarantee against converging to a local minimum when jointly tuning both the n-gram weighting and interpolation parameters, we have found that initializing the parameters to zero generally yields good performance.

4 Experiments

4.1 Setup

In this work, we evaluate the perplexity and WER of various trigram LMs trained with n-gram weighting on a lecture transcription task (Glass et al., 2007). The target data consists of 20 lectures from an introductory computer science course, from which we withhold the first 10 lectures for the development

[1] HMM-LDA is performed using 20 states and 50 topics with a 3rd-order HMM. Hyperparameters are sampled with a log-normal Metropolis proposal. The model with the highest likelihood from among 10,000 iterations of Gibbs sampling is used.

Dataset	# Words	# Sents	# Docs
Textbook	131,280	6,762	271
Lectures	1,994,225	128,895	230
Switchboard	3,162,544	262,744	4,876
CS Dev	93,353	4,126	10
CS Test	87,527	3,611	10

Table 2: Summary of evaluation corpora.

	Perplexity		WER	
Model	Dev	Test	Dev	Test
FixKN(1)	174.7	196.7	34.9%	36.8%
+ W(h^{doc})	172.9	194.8	34.7%	36.7%
FixKN(3)	168.6	189.3	34.9%	36.9%
+ W(h^{doc})	166.8	187.8	34.6%	36.6%
FixKN(10)	167.5	187.6	35.0%	37.2%
+ W(h^{doc})	165.3	185.8	34.7%	36.8%
KN(1)	169.7	190.4	35.0%	37.0%
+ W(h^{doc})	167.3	188.2	34.8%	36.7%
KN(3)	163.4	183.1	35.0%	37.1%
+ W(h^{doc})	161.1	181.2	34.7%	36.8%
KN(10)	162.3	181.8	35.1%	37.1%
+ W(h^{doc})	160.1	180.0	34.8%	36.8%

Table 3: Performance of n-gram weighting with a variety of Kneser-Ney settings. FixKN(d): Kneser-Ney with d fixed discount parameters. KN(d): FixKN(d) with tuned values. W(*feat*): n-gram weighting with *feat* feature.

set (CS Dev) and use the last 10 for the test set (CS Test). For training, we will consider the course textbook with topic-specific vocabulary (Textbook), numerous high-fidelity transcripts from a variety of general seminars and lectures (Lectures), and the out-of-domain LDC Switchboard corpus of spontaneous conversational speech (Switchboard) (Godfrey and Holliman, 1993). Table 2 summarizes all the evaluation data.

To compute the word error rate, we use a speaker-independent speech recognizer (Glass, 2003) with a large-margin discriminative acoustic model (Chang, 2008). The lectures are pre-segmented into utterances via forced alignment against the reference transcripts (Hazen, 2006). Since all the models considered in this work can be encoded as n-gram backoff models, they are applied directly during the first recognition pass instead of through a subsequent n-best rescoring step.

Model	Perplexity	WER
Lectures	189.3	36.9%
+ W(h^{doc})	187.8 (-0.8%)	36.6%
Textbook	326.1	43.1%
+ W(h^{doc})	317.5 (-2.6%)	43.1%
LI(Lectures + Textbook)	141.6	33.7%
+ W(h^{doc})	136.6 (-3.5%)	32.7%

Table 4: N-gram weighting with linear interpolation.

4.2 Smoothing

In Table 3, we compare the performance of n-gram weighting with the h^{doc} document entropy feature for various modified Kneser-Ney smoothing configurations (Chen and Goodman, 1998) on the Lectures dataset. Specifically, we considered varying the number of discount parameters per n-gram order from 1 to 10. The original and modified Kneser-Ney smoothing algorithms correspond to a setting of 1 and 3, respectively. Furthermore, we explored using both fixed parameter values estimated from n-gram count statistics and tuned values that minimize the development set perplexity.

In this task, while the test set perplexity tracks the development set perplexity well, the WER correlates surprisingly poorly with the perplexity on both the development and test sets. Nevertheless, n-gram weighting consistently reduces the absolute test set WER by a statistically significant average of 0.3%, according to the Matched Pairs Sentence Segment Word Error test (Pallet et al., 1990). Given that we obtained the lowest development set WER with the fixed 3-parameter modified Kneser-Ney smoothing, all subsequent experiments are conducted using this smoothing configuration.

4.3 Linear Interpolation

Applied to the Lectures dataset in isolation, n-gram weighting with the h^{doc} feature reduces the test set WER by 0.3% by de-emphasizing the probability contributions from off-topic n-grams and shifting their weights to the backoff distributions. Ideally though, such weights should be distributed to on-topic n-grams, perhaps from other LM components.

In Table 4, we present the performance of applying n-gram weighting to the Lectures and Textbook models individually versus in combination via linear interpolation (LI), where we optimize the n-gram

Model	Perplexity	WER
LI(Lectures + Textbook)	141.6	33.7%
+ W(*Random*)	141.5 (-0.0%)	33.7%
+ W($\log(c)$)	137.5 (-2.9%)	32.8%
+ W(f^{doc})	136.3 (-3.7%)	32.8%
+ W(f^{course})	136.5 (-3.6%)	32.7%
+ W(f^{speaker})	138.1 (-2.5%)	33.0%
+ W(h^{doc})	136.6 (-3.5%)	**32.7%**
+ W(h^{course}) ★	136.1 (-3.9%)	$\underline{32.7\%}$
+ W(h^{speaker})	138.6 (-2.1%)	33.1%
+ W(t^{doc})	$\underline{\textbf{134.8}}$ (-4.8%)	33.2%
+ W(t^{course})	136.4 (-3.6%)	33.1%
+ W(t^{speaker})	136.4 (-3.7%)	33.2%

Table 5: N-gram weighting with various features.

weighting and interpolation parameters jointly. The interpolated model with n-gram weighting achieves perplexity improvements roughly additive of the reductions obtained with the individual models. However, the 1.0% WER drop for the interpolated model significantly exceeds the sum of the individual reductions. Thus, as we will examine in more detail in Section 5.1, n-gram weighting allows probabilities to be shifted from less relevant n-grams in one component to more specific n-grams in another.

4.4 Features

With n-gram weighting, we can model the weighting function $\beta(hw)$ as a log-linear combination of any n-gram features. In Table 5, we show the effect various features have on the performance of linearly interpolating Lectures and Textbook. As the documents from the Lectures dataset is annotated with course and speaker metadata attributes, we include the n-gram frequency f, entropy h, and topic probability t features computed from the lectures grouped by the 16 unique courses and 299 unique speakers.[2]

In terms of perplexity, the use of the *Random* feature has negligible impact on the test set performance, as expected. On the other hand, the $\log(c)$ count feature reduces the perplexity by nearly 3%, as it correlates with the generality of the n-grams. By using features that leverage the information from document segmentation and associated metadata, we are generally able to achieve further perplexity reductions. Overall, the frequency and entropy features perform roughly equally. However, by considering information from the more sophisticated HMM-LDA topic model, the topic probability feature t^{doc} achieves significantly lower perplexity than any other feature in isolation.

In terms of WER, the *Random* feature again shows no effect on the baseline WER of 33.7%. However, to our surprise, the use of the simple $\log(c)$ feature achieves nearly the same WER improvement as the best segmentation-based feature, whereas the more sophisticated features computed from HMM-LDA labels only obtain half of the reduction even though they have the best perplexities.

When comparing the performance of different n-gram weighting features on this data set, the perplexity correlates poorly with the WER, on both the development and test sets. Fortunately, the features that yield the lowest perplexity and WER on the development set also yield one of the lowest perplexities and WERs, respectively, on the test set. Thus, during feature selection for speech recognition applications, we should consider the development set WER. Specifically, since the differences in WER are often statistically insignificant, we will select the feature that minimizes the sum of the development set WER and log perplexity, or cross-entropy.[3]

In Tables 5 and 6, we have underlined the perplexities and WERs of the features with the lowest corresponding development set values (not shown) and bolded the lowest test set values. The features that achieve the lowest combined cross-entropy and WER on the development set are starred.

4.5 Feature Combination

Unlike most previous work, n-gram weighting enables a systematic integration of features computed from multiple document partitions. In Table 6, we compare the performance of various feature combinations. We experiment with incrementally adding features that yield the lowest combined development set cross-entropy and WER. Overall, this metric appears to better predict the test set WER than either the development set perplexity or WER alone.

[2] Features that are not applicable to a particular corpus (e.g. h^{course} for Textbook) are removed from the n-gram weighting computation for that component. Thus, models with course and speaker features have fewer tunable parameters than the others.

[3] The choice of cross-entropy instead of perplexity is partially motivated by the linear correlation reported by (Chen and Goodman, 1998) between cross-entropy and WER.

Features	Perplexity	WER
h^{course}	136.1	32.7%
+ $\log(c)$	135.4 (-0.5%)	32.6%
+ f^{doc}	135.1 (-0.7%)	32.6%
+ h^{doc}	135.6 (-0.5%)	32.6%
+ t^{doc} ★	**133.2** (-2.1%)	**32.6%**
+ f^{course}	136.0 (-0.1%)	32.6%
+ t^{course}	134.8 (-1.0%)	32.9%
+ f^{speaker}	136.0 (-0.1%)	32.6%
+ h^{speaker}	136.1 (-0.0%)	32.8%
+ t^{speaker}	134.7 (-1.0%)	32.7%
h^{course} + t^{doc}	133.2	32.6%
+ $\log(c)$	132.8 (-0.3%)	32.5%
+ f^{doc} ★	**132.8** (-0.4%)	**32.5%**
+ h^{doc}	133.0 (-0.2%)	32.5%
+ f^{course}	133.1 (-0.1%)	32.5%
+ t^{course}	133.0 (-0.1%)	32.6%
+ f^{speaker}	133.1 (-0.1%)	32.5%
+ h^{speaker}	133.2 (-0.0%)	32.6%
+ t^{speaker}	133.1 (-0.1%)	32.7%

Table 6: N-gram weighting with feature combinations.

Using the combined feature selection technique, we notice that the greedily selected features tend to differ in the choice of document segmentation and feature type, suggesting that n-gram weighting can effectively integrate the information provided by the document metadata. By combining features, we are able to further reduce the test set WER by a statistically significant ($p < 0.001$) 0.2% over the best single feature model.

4.6 Advanced Interpolation

While n-gram weighting with all three features is able to reduce the test set WER by 1.2% over the linear interpolation baseline, linear interpolation is not a particularly effective interpolation technique. In Table 7, we compare the effectiveness of n-gram weighting in combination with better interpolation techniques, such as count merging (CM) (Bacchiani et al., 2006) and generalized linear interpolation (GLI) (Hsu, 2007). As expected, the use of more sophisticated interpolation techniques decreases the perplexity and WER reductions achieved by n-gram weighting by roughly half for a variety of feature combinations. However, all improvements remain statistically significant.

Model	Perplexity	WER
Linear(L + T)	141.6	33.7%
+ W(h^{course})	136.1 (-3.9%)	32.7%
+ W(t^{doc})	133.2 (-5.9%)	32.6%
+ W(f^{doc})	132.8 (-6.2%)	32.5%
CM(L + T)	137.9	33.0%
+ W(h^{course})	135.5 (-1.8%)	32.4%
+ W(t^{doc})	133.4 (-3.3%)	32.4%
+ W(f^{doc})	133.2 (-3.5%)	32.4%
$\text{GLI}_{\log(1+\tilde{c})}$(L + T)	135.9	33.0%
+ W(h^{course})	133.0 (-2.2%)	32.4%
+ W(t^{doc})	130.6 (-3.9%)	32.4%
+ W(f^{doc})	130.5 (-4.2%)	32.4%

Table 7: Effect of interpolation technique. L: Lectures, T: Textbook.

Feature	Parameter Values
h^{doc}	$\boldsymbol{\theta}^{\mathsf{L}} = [3.42, 1.46, 0.12]$
	$\boldsymbol{\theta}^{\mathsf{T}} = [-0.45, -0.35, -0.73]$
	$[\lambda^{\mathsf{L}}, \lambda^{\mathsf{T}}] = [0.67, 0.33]$
t^{doc}	$\boldsymbol{\theta}^{\mathsf{L}} = [-2.33, -1.63, -1.19]$
	$\boldsymbol{\theta}^{\mathsf{T}} = [1.05, 0.46, 0.12]$
	$[\lambda^{\mathsf{L}}, \lambda^{\mathsf{T}}] = [0.68, 0.32]$

Table 8: N-gram weighting parameter values. $\boldsymbol{\theta}^{\mathsf{L}}$, $\boldsymbol{\theta}^{\mathsf{T}}$: parameters for each order of the Lectures and Textbook trigram models, λ^{L},λ^{T}: linear interpolation weights.

Although the WER reductions from better interpolation techniques are initially statistically significant, as we add features to n-gram weighting, the differences among the interpolation methods shrink significantly. With all three features combined, the test set WER difference between linear interpolation and generalized linear interpolation loses its statistical significance. In fact, we can obtain statistically the same WER of 32.4% using the simpler model of count merging and n-gram weighting with h^{course}.

5 Analysis

5.1 Weighting Parameters

To obtain further insight into how n-gram weighting improves the resulting n-gram model, we present in Table 8 the optimized parameter values for the linear interpolation model between Lectures and Textbook using n-gram weighting with h^{doc} and t^{doc} features. Using $\beta(hw) = \exp(\boldsymbol{\Phi}(hw) \cdot \boldsymbol{\theta})$ to model the n-gram weights, a positive value of θ_i corresponds to

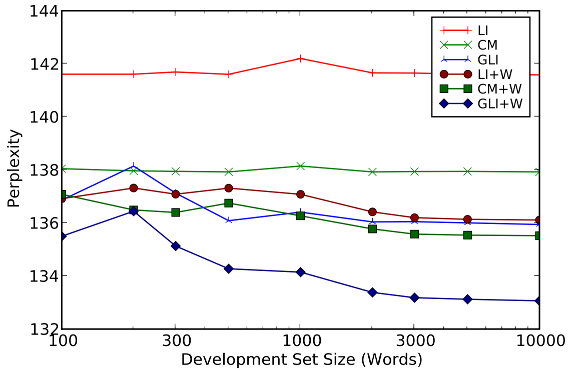

Figure 1: Test set perplexity vs. development set size.

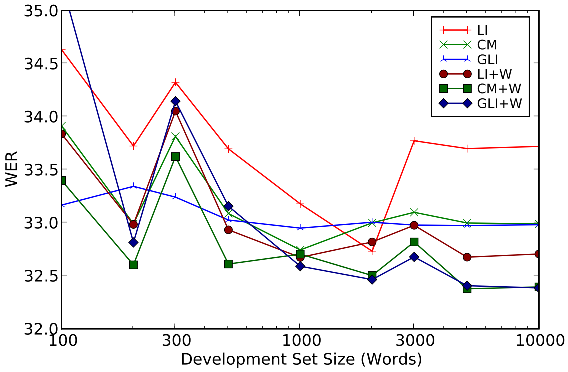

Figure 2: Test set WER vs. development set size.

increasing the weights of the i^{th} order n-grams with positive feature values.

For the h^{doc} normalized entropy feature, values close to 1 correspond to n-grams that are evenly distributed across the documents. When interpolating Lectures and Textbook, we obtain consistently positive values for the Lectures component, indicating a de-emphasis on document-specific terms that are unlikely to be found in the target computer science domain. On the other hand, the values corresponding to the Textbook component are consistently negative, suggesting a reduced weight for mismatched style terms that appear uniformly across textbook sections.

For t^{doc}, values close to 1 correspond to n-grams ending frequently on topic words with uneven distribution across documents. Thus, as expected, the signs of the optimized parameter values are flipped. By de-emphasizing topic n-grams from off-topic components and style n-grams from off-style components, n-gram weighting effectively improves the performance of the resulting language model.

5.2 Development Set Size

So far, we have assumed the availability of a large development set for parameter tuning. To obtain a sense of how n-gram weighting performs with smaller development sets, we randomly select utterances from the full development set and plot the test set perplexity in Figure 1 as a function of the development set size for various modeling techniques.

As expected, GLI outperforms both LI and CM. However, whereas LI and CM essentially converge in test set perplexity with only 100 words of development data, it takes about 500 words before GLI converges due to the increased number of parameters. By adding n-gram weighting with the h^{course} feature, we see a significant drop in perplexity for all models at all development set sizes. However, the performance does not fully converge until 3,000 words of development set data.

As shown in Figure 2, the test set WER behaves more erratically, as the parameters are tuned to minimize the development set perplexity. Overall, n-gram weighting decreases the WER significantly, except when applied to GLI with less than 1000 words of development data when the perplexity of GLI has not itself converged. In that range, CM with n-gram weighting performs the best. However, with more development data, GLI with n-gram weighting generally performs slightly better. From these results, we conclude that although n-gram weighting increases the number of tuning parameters, they are effective in improving the test set performance even with only 100 words of development set data.

5.3 Training Set Size

To characterize the effectiveness of n-gram weighting as a function of the training set size, we evaluate the performance of various interpolated models with increasing subsets of the Lectures corpus and the full Textbook corpus. Overall, every doubling of the number of training set documents decreases both the test set perplexity and WER by approximately 7 points and 0.8%, respectively. To better compare results, we plot the performance difference between various models and linear interpolation in Figures 3 and 4.

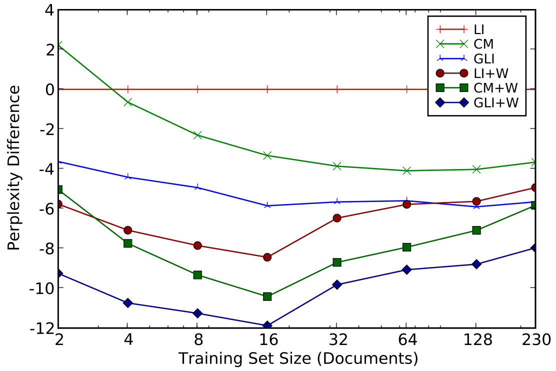

Figure 3: Test set perplexity vs. training set size.

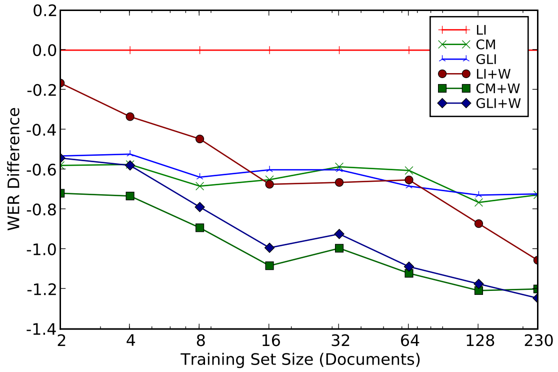

Figure 4: Test set WER vs. training set size.

Interestingly, the peak gain obtained from n-gram weighting with the h^{doc} feature appears at around 16 documents for all interpolation techniques. We suspect that as the number of documents initially increases, the estimation of the h^{doc} features improves, resulting in larger perplexity reduction from n-gram weighting. However, as the diversity of the training set documents increases beyond a certain threshold, we experience less document-level sparsity. Thus, we see decreasing gain from n-gram weighting beyond 16 documents.

For all interpolation techniques, even though the perplexity improvements from n-gram weighting decrease with more documents, the WER reductions actually increase. N-gram weighting showed statistically significant reductions for all configurations except generalized linear interpolation with less than 8 documents. Although count merging with n-gram weighting has the lowest WER for most training set sizes, GLI ultimately achieves the best test set WER with the full training set.

5.4 Training Corpora

In Table 9, we compare the performance of n-gram weighting with different combination of training corpora and interpolation techniques to determine its effectiveness across different training conditions. With the exception of interpolating Lectures and Switchboard using count merging, all other model combinations yield statistically significant improvements with n-gram weighting using h^{course}, t^{doc}, and f^{doc} features.

The results suggest that n-gram weighting with these features is most effective when interpolating corpora that differ in how they match the target domain. Whereas the Textbook corpus is the only corpus with matching topic, both Lectures and Switchboard have a similar matching spoken conversational style. Thus, we see the least benefit from n-gram weighting when interpolating Lectures and Switchboard. By combining Lectures, Textbook, and Switchboard using generalized linear interpolation with n-gram weighting using h^{course}, t^{doc}, and f^{doc} features, we achieve our best test set WER of 32.2% on the lecture transcription task, a full 1.5% over the initial linear interpolation baseline.

Model	L + T	L + S	T + S	L + T + S
LI	33.7%	36.7%	36.4%	33.6%
LI + W	32.5%	36.4%	35.7%	32.5%
CM	33.0%	36.6%	35.5%	32.9%
CM + W	32.4%	36.5%	35.4%	32.3%
GLI	33.0%	36.6%	35.7%	32.8%
GLI + W	32.4%	36.4%	35.3%	32.2%

Table 9: Test set WER with various training corpus combinations. L: Lectures, T: Textbook, S: Switchboard, W: n-gram weighting.

6 Conclusion & Future Work

In this work, we presented the n-gram weighting technique for adjusting the probabilities of n-grams according to a set of features. By utilizing features derived from the document segmentation and associated metadata inherent in many training corpora, we achieved up to a 1.2% and 0.6% WER reduction over the linear interpolation and count merging baselines, respectively, using n-gram weighting on a lecture transcription task.

We examined the performance of various n-gram weighting features and generally found entropy-based features to offer the best predictive performance. Although the topic probability features derived from HMM-LDA labels yield additional improvements when applied in combination with the normalized entropy features, the computational cost of performing HMM-LDA may not justify the marginal benefit in all scenarios.

In situations where the document boundaries are unavailable or when finer segmentation is desired, automatic techniques for document segmentation may be applied (Malioutov and Barzilay, 2006). Synthetic metadata information may also be obtained via clustering techniques (Steinbach et al., 2000). Although we have primarily focused on n-gram weighting features derived from segmentation information, it is also possible to consider other features that correlate with n-gram relevance.

N-gram weighting and other approaches to cross-domain language modeling require a matched development set for model parameter tuning. Thus, for future work, we plan to investigate the use of the initial recognition hypotheses as the development set, as well as manually transcribing a subset of the test set utterances.

As speech and natural language applications shift towards novel domains with limited matched training data, better techniques are needed to maximally utilize the often abundant partially matched data. In this work, we examined the effectiveness of the n-gram weighting technique for estimating language models in these situations. With similar investments in acoustic modeling and other areas of natural language processing, we look forward to an ever increasing diversity of practical speech and natural language applications.

Availability An implementation of the n-gram weighting algorithm is available in the MIT Language Modeling (MITLM) toolkit (Hsu and Glass, 2008): http://www.sls.csail.mit.edu/mitlm/.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive feedback. This research is supported in part by the T-Party Project, a joint research program between MIT and Quanta Computer Inc.

References

Michiel Bacchiani, Michael Riley, Brian Roark, and Richard Sproat. 2006. MAP adaptation of stochastic grammars. *Computer Speech & Language*, 20(1):41–68.

Jerome R. Bellegarda. 2004. Statistical language model adaptation: Review and perspectives. *Speech Communication*, 42(1):93–108.

Hung-An Chang. 2008. Large margin Gaussian mixture modeling for automatic speech recognition. Massachusetts Institute of Technology. Masters Thesis.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. In *Technical Report TR-10-98*. Computer Science Group, Harvard University.

Jianfeng Gao and Kai-Fu Lee. 2000. Distribution-based pruning of backoff language models. In *Proc. Association of Computational Linguistics*, pages 579–588, Hong Kong, China.

James Glass, Timothy J. Hazen, Lee Hetherington, and Chao Wang. 2004. Analysis and processing of lecture audio data: Preliminary investigations. In *Proc. HLT-NAACL Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval*, pages 9–12, Boston, MA, USA.

James Glass, Timothy J. Hazen, Scott Cyphers, Igor Malioutov, David Huynh, and Regina Barzilay. 2007. Recent progress in the MIT spoken lecture processing project. In *Proc. Interspeech*, pages 2553–2556, Antwerp, Belgium.

James Glass. 2003. A probabilistic framework for segment-based speech recognition. *Computer Speech & Language*, 17(2-3):137–152.

John J. Godfrey and Ed Holliman. 1993. Switchboard-1 transcripts. Linguistic Data Consortium, Philadelphia, PA, USA.

Thomas L. Griffiths, Mark Steyvers, David M. Blei, and Joshua B. Tenenbaum. 2005. Integrating topics and syntax. In *Advances in Neural Information Processing Systems 17*, pages 537–544. MIT Press, Cambridge, MA, USA.

T.J. Hazen. 2006. Automatic alignment and error correction of human generated transcripts for long speech recordings. In *Proc. Interspeech*, Pittsburgh, PA, USA.

Bo-June (Paul) Hsu and James Glass. 2006. Style & topic language model adaptation using HMM-LDA. In *Proc. Empirical Methods in Natural Language Processing*, pages 373–381, Sydney, Australia.

Bo-June (Paul) Hsu and James Glass. 2008. Iterative language model estimation: Efficient data structure & algorithms. In *Proc. Interspeech*, Brisbane, Australia.

Bo-June (Paul) Hsu. 2007. Generalized linear interpolation of language models. In *Proc. Automatic Speech Recognition and Understanding*, pages 136–140, Kyoto, Japan.

Igor Malioutov and Regina Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *Proc. Association for Computational Linguistics*, pages 25–32, Sydney, Australia.

D. Pallet, W. Fisher, and Fiscus. 1990. Tools for the analysis of benchmark speech recognition tests. In *Proc. ICASSP*, Albuquerque, NM, USA.

William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. 2007. *Numerical Recipes*. Cambridge University Press, 3rd edition.

Michael Steinbach, George Karypis, and Vipin Kumar. 2000. A comparison of document clustering techniques. Technical Report #00-034, University of Minnesota.