Style & Topic Language Model Adaptation Using HMM-LDA

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

Abstract

Adapting language models across styles and topics, such as for lecture transcription, involves combining generic style models with topic-specific content relevant to the target document. In this work, we investigate the use of the Hidden Markov Model with Latent Dirichlet Allocation (HMM-LDA) to obtain syntactic state and semantic topic assignments to word instances in the training corpus. From these context-dependent labels, we construct style and topic models that better model the target document, and extend the traditional bag-of-words topic models to n-grams. Experiments with static model interpolation yielded a perplexity and relative word error rate (WER) reduction of 7.1% and 2.1%, respectively, over an adapted trigram baseline. Adaptive interpolation of mixture components further reduced perplexity by 9.5% and WER by a modest 0.3%.

1 Introduction

With the rapid growth of audio-visual materials available over the web, effective language modeling of the diverse content, both in style and topic, becomes essential for efficient access and management of this information. As a prime example, successful language modeling for academic lectures not only enables the initial transcription via automatic speech recognition, but also assists educators and students in the creation and navigation of these materials through annotation, retrieval, summarization, and even translation of the embedded content.

Compared with other types of audio content, lecture speech often exhibits a high degree of spontaneity and focuses on narrow topics with specific terminology (Furui, 2003; Glass et al, 2004). Unfortunately, training corpora available for language modeling rarely match the target lecture in both style and topic. While transcripts from other lectures better match the style of the target lecture than written text, it is often difficult to find transcripts on the target topic. On the other hand, although topic-specific vocabulary can be gleaned from related text materials, such as the textbook and lecture slides, written language is a poor predictor of how words are actually spoken. Furthermore, given that the precise topic of a target lecture is often unknown a priori and may even shift over time, it is generally difficult to identify topically related documents. Thus, an effective language model (LM) need to not only account for the casual speaking style of lectures, but also accommodate the topic-specific vocabulary of the subject matter. Moreover, the ability of the language model to dynamically adapt over the course of the lecture could prove extremely useful for both increasing transcription accuracy, as well as providing evidence for lecture segmentation and information retrieval.

In this paper, we investigate the application of the syntactic state and semantic topic assignments from the Hidden Markov Model with Latent Dirichlet Allocation model to the problem of language modeling. We explore the use of these context-dependent labels to identify style and learn topics from both a large number of spoken lectures as well as written text. By dynamically interpolating lecture style models with topic-specific models, we obtain language models that better describe the subtopic structure within a lecture. Initial experiments demonstrate a 16.1% perplexity reduction and a 2.4% WER reduction over an adapted trigram baseline.

Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pages 373–381,
Sydney, July 2006. ©2006 Association for Computational Linguistics

In the following sections, we first summarize related research on adaptive and topic-mixture language models, and describe previous work on the HMM-LDA model. We then examine the ability of the model to learn syntactic classes as well as topics from textbook materials and lecture transcripts. Next, we describe a variety of language model experiments we performed to combine style and topic models constructed from the state and topic labels with conventional trigram models trained from both spoken and written materials. We also demonstrate the use of the combined model in an on-line adaptive mode. Finally, we summarize the results of this research and suggest future opportunities for related modeling techniques in spoken lecture and other content processing research.

2 Adaptive and Topic-Mixture LMs

The concept of adaptive and topic-mixture language models has been previously explored by many researchers. Adaptive language modeling exploits the property that words appearing earlier in a document are likely to appear again. Cache language models (Kuhn and De Mori, 1990; Clarkson and Robinson, 1997) leverage this observation and increase the probability of previously observed words in a document when predicting the next word. By interpolating with a conditional trigram cache model, Goodman (2001) demonstrated up to 34% decrease in perplexity over a trigram baseline for small training sets.

The cache intuition has been extended by attempting to increase the probability of unobserved but topically related words. Specifically, given a mixture model with topic-specific components, we can increase the mixture weights of the topics corresponding to previously observed words to better predict the next word. Some of the early work in this area used a maximum entropy language model framework to trigger increases in likelihood of related words (Lau et al., 1993; Rosenfeld, 1996).

A variety of methods has been used to explore topic-mixture models. To model a mixture of topics within a document, the sentence mixture model (Iyer and Ostendorf, 1999) builds multiple topic models from clusters of training sentences and defines the probability of a target sentence as a weighted combination of its probability under each topic model. Latent Semantic Analysis (LSA) has been used to cluster topically related words and has demonstrated significant reduction in perplexity and word error rate (Bellegarda, 2000). Probabilistic LSA (PLSA) has been used to decompose documents into component word distributions and create unigram topic models from these distributions. Gildea and Hofmann (1999) demonstrated noticeable perplexity reduction via dynamic combination of these unigram topic models with a generic trigram model.

To identify topics from an unlabeled corpus, (Blei et al., 2003) extends PLSA with the Latent Dirichlet Allocation (LDA) model that describes each document in a corpus as generated from a mixture of topics, each characterized by a word unigram distribution. Hidden Markov Model with LDA (HMM-LDA) (Griffiths et al., 2004) further extends this topic mixture model to separate syntactic words from content words whose distributions depend primarily on local context and document topic, respectively.

In the specific area of lecture processing, previous work in language model adaptation has primarily focused on customizing a fixed n-gram language model for each lecture by combining n-gram statistics from general conversational speech, other lectures, textbooks, and other resources related to the target lecture (Nanjo and Kawahara, 2002, 2004; Leeuwis et al., 2003; Park et al., 2005).

Most of the previous work on topic-mixture models focuses on in-domain adaptation using large amounts of matched training data. However, most, if not all, of the data available to train a lecture language model are either cross-domain or cross-style. Furthermore, although adaptive models have been shown to yield significant perplexity reduction on clean transcripts, the improvements tend to diminish when working with speech recognizer hypotheses with high WER.

In this work, we apply the concept of dynamic topic adaptation to the lecture transcription task. Unlike previous work, we first construct a style model and a topic-domain model using the classification of word instances into syntactic states and topics provided by HMM-LDA. Furthermore, we leverage the context-dependent labels to extend topic models from unigrams to n-grams, allowing for better prediction of transitions involving topic words. Note that although this work focuses on the use of HMM-LDA to generate the state and topic labels, any method that yields such labels suffices for the purpose of the language modeling experiments. The following section describes the HMM-LDA framework in more detail.

3 HMM-LDA

3.1 Latent Dirichlet Allocation

Discrete Principal Component Analysis describes a family of models that decompose a set of feature vectors into its principal components (Buntine and Jakulin, 2005). Describing feature vectors via their components reduces the number of parameters required to model the data, hence improving the quality of the estimated parameters when given limited training data. LSA, PLSA, and LDA are all examples from this family.

Given a predefined number of desired components, LSA models feature vectors by finding a set of orthonormal components that maximize the variance using singular value decomposition (Deerwester et al., 1990). Unfortunately, the component vectors may contain non-interpretable negative values when working with word occurrence counts as feature vectors. PLSA eliminates this problem by using non-negative matrix factorization to model each document as a weighted combination of a set of non-negative feature vectors (Hofmann, 1999). However, because the number of parameters grows linearly with the number of documents, the model is prone to overfitting. Furthermore, because each training document has its own set of topic weight parameters, PLSA does not provide a generative framework for describing the probability of an unseen document (Blei et al., 2003).

To address the shortcomings of PLSA, Blei et al. (2003) introduced the LDA model, which further imposes a Dirichlet distribution on the topic mixture weights corresponding to the documents in the corpus. With the number of model parameters dependent only on the number of topic mixtures and vocabulary size, LDA is less prone to overfitting and is capable of estimating the probability of unobserved test documents.

Empirically, LDA has been shown to outperform PLSA in corpus perplexity, collaborative filtering, and text classification experiments (Blei et al., 2003). Various extensions to the basic LDA model have since been proposed. The Author Topic model adds an additional dependency on the author(s) to the topic mixture weights of each document (Rosen-Zvi et al., 2005). The Hierarchical Dirichlet Process is a nonparametric model that generalizes distribution parameter modeling to multiple levels. Without having to estimate the number of mixture components, this model has been shown to match the best result from LDA on a document modeling task (Teh et al., 2004).

3.2 Hidden Markov Model with LDA

HMM-LDA model proposed by Griffiths et al. (2004) combines the HMM and LDA models to separate syntactic words with local dependencies from topic-dependent content words without requiring any labeled data. Similar to HMM-based part-of-speech taggers, HMM-LDA maps each word in the document to a hidden syntactic state. Each state generates words according to a unigram distribution except the special topic state, where words are modeled by document-specific mixtures of topic distributions, as in LDA. Figure 1 describes this generative process in more detail.

For each document d in the corpus:

1. Draw topic weights θ^d from Dirichlet(α)
2. For each word w_i in document d:
   a. Draw topic z_i from Multinomial(θ^d)
   b. Draw state s_i from Multinomial($\pi^{s_{i-1}}$)
   c. Draw word w_i from:

$$\begin{cases} \text{Multinomial}(\beta^{z_i}) & s_i = \text{s}_{\text{topic}} \\ \text{Multinomial}(\gamma^{s_i}) & \textit{otherwise} \end{cases}$$

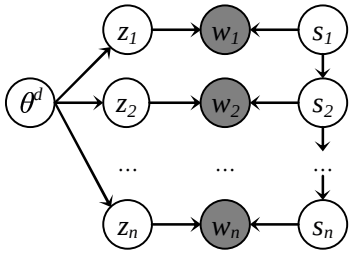

Figure 1: Generative framework and graphical model representation of HMM-LDA. The number of states and topics are pre-specified. The topic mixture for each document is modeled with a Dirichlet distribution. Each word w_i in the n-word document is generated from its hidden state s_i or hidden topic z_i if s_i is the special topic state.

Unlike vocabulary selection techniques that separate domain-independent words from topic-specific keywords using word collocation statistics, HMM-LDA classifies each word instance according to its context. Thus, an instance of the word "*return*" may be assigned to a syntactic state in "to *return* a", but classified as a topic keyword in "expected *return* for". By labeling each word in the training set with its syntactic state and mixture topic, HMM-LDA not only separates stylistic words from content words in a context-dependent manner, but also decomposes the corpus into a set of topic word distributions. This form of soft, context-dependent classifica-

tion has many potential uses for language modeling, topic segmentation, and indexing.

3.3 Training

To train an HMM-LDA model, we employ the MATLAB Topic Modeling Toolbox 1.3 (Griffiths and Steyvers, 2004; Griffiths et al., 2004). This particular implementation performs Gibbs sampling, a form of Markov chain Monte Carlo (MCMC), to estimate the optimal model parameters fitted to the training data. Specifically, the algorithm creates a Markov chain whose stationary distribution matches the expected distribution of the state and topic labels for each word in the training corpus. Starting from random labels, Gibbs sampling sequentially samples the label for each hidden variable conditioned on the current value of all other variables. After a sufficient number of iterations, the Markov chain converges to the stationary distribution. We can easily compute the posterior word distribution for each state and topic from a single sample by averaging over the label counts and prior parameters. With a sufficiently large training set, we will have enough words assigned to each state and topic to yield a reasonable approximation to the underlying distribution.

In the following sections, we examine the application of models derived from the HMM-LDA labels to the task of spoken lecture transcription and explore techniques on adaptive topic modeling to construct a better lecture language model.

4 HMM-LDA Analysis

Our language modeling experiments have been conducted on high-fidelity transcripts of approximately 168 hours of lectures from three undergraduate subjects in math, physics, and computer science (CS), as well as 79 seminars covering a wide range of topics (Glass et al., 2004). For evaluation, we withheld the set of 20 CS lectures and used the first 10 lectures as a development set and the last 10 lectures for the test set. The remainder of these data was used for training and will be referred to as the *Lectures* dataset.

To supplement the out-of-domain lecture transcripts with topic-specific textual resources, we added the CS course textbook (*Textbook*) as additional training data for learning the target topics. To create topic-cohesive documents, the textbook is divided at every section heading to form 271 documents. Next, the text is heuristically segmented at sentence-like boundaries and normalized into the words corresponding to the spoken form of the text. Table 1 summarizes the data used in this evaluation.

Dataset	Documents	Sentences	Vocabulary	Words
Lectures	150	58,626	25,654	1,390,039
Textbook	271	6,762	4,686	131,280
CS Dev	10	4,102	3,285	93,348
CS Test	10	3,595	3,357	87,518

Table 1: Summary of evaluation datasets.

In the following analysis, we ran the Gibbs sampler against the *Lectures* dataset for a total of 2800 iterations, computing a model every 10 iterations, and took the model with the lowest perplexity as the final model. We built the model with 20 states and 100 topics based on preliminary experiments. We also trained an HMM-LDA model on the *Textbook* dataset using the same model parameters. We ran the sampler for a total of 2000 iterations, computing the perplexity every 100 iterations. Again, we selected the lowest perplexity model as the final model.

4.1 Semantic Topics

HMM-LDA extracts words whose distributions vary across documents and clusters them into a set of components. In Figure 2, we list the top 10 words from a random selection of 10 topics computed from the *Lectures* dataset. As shown, the words assigned to the LDA topic state are representative of content words and are grouped into broad semantic topics. For example, topic 4, 8, and 9 correspond to machine learning, linear algebra, and magnetism, respectively.

Since the *Lectures* dataset consists of speech transcripts with disfluencies, it is interesting to

1	2	3	4	5	6	7	8	9	10
center	work	rights	system	<laugh>	<partial>	class	basis	magnetic	light
world	research	human	things	her	memory	people	v	current	red
and	right	U.	robot	children	ah	tax	<eh>	field	water
ideas	people	S.	systems	book	brain	wealth	vector	loop	colors
new	computing	government	work	Cambridge	animal	social	matrix	surface	white
technology	network	international	example	books	okay	American	transformation	direction	angle
innovation	system	countries	person	street	eye	power	linear	e	blue
community	information	president	robots	city	synaptic	world	eight	law	here
place	software	world	learning	library	receptors	<unintelligible>	output	flux	rainbow
building	computers	support	machine	brother	mouse	society	t	m	sun

Figure 2: The top 10 words from 10 randomly selected topics computed from the *Lectures* dataset.

observe that "<*laugh*>" is the top word in a topic corresponding to childhood memories. Cursory examination of the data suggests that the speakers talking about children tend to laugh more during the lecture. Although it may not be desirable to capture speaker idiosyncrasies in the topic mixtures, HMM-LDA has clearly demonstrated its ability to capture distinctive semantic topics in a corpus. By leveraging all documents in the corpus, the model yields smoother topic word distributions that are less vulnerable to overfitting.

Since HMM-LDA labels the state and topic of each word in the training corpus, we can also visualize the results by color-coding the words by their topic assignments. Figure 3 shows a color-coded excerpt from a topically coherent paragraph in the *Textbook* dataset. Notice how most of the content words (uppercase) are assigned to the same topic/color. Furthermore, of the 7 instances of the words "*and*" and "*or*" (underlined), 6 are correctly classified as syntactic or topic words, demonstrating the context-dependent labeling capabilities of the HMM-LDA model. Moreover, from these labels, we can identify multi-word topic key phrases (e.g. *output signals*, *input signal*, "*and*" *gate*) in addition to standalone keywords, an observation we will leverage later on with n-gram topic models.

We draw an INVERTER SYMBOLICALLY as in Figure 3.24. An AND GATE, also shown in Figure 3.24, is a PRIMITIVE FUNCTION box with two INPUTS and ONE OUTPUT. It drives its OUTPUT SIGNAL to a value that is the LOGICAL AND of the INPUTS. That is, if both of its INPUT SIGNALS BECOME 1. Then ONE and GATE DELAY time later the AND GATE will force its OUTPUT SIGNAL TO be 1; otherwise the OUTPUT will be 0. An OR GATE is a SIMILAR two INPUT PRIMITIVE FUNCTION box that drives its OUTPUT SIGNAL to a value that is the LOGICAL OR of the INPUTS. That is, the OUTPUT will BECOME 1 if at least ONE of the INPUT SIGNALS is 1; otherwise the OUTPUT will BECOME 0.

Figure 3: Color-coded excerpt from the *Textbook* dataset showing the context-dependent topic labels. Syntactic words appear black in lowercase. Topic words are shown in uppercase with their respective topic colors. All instances of the words "*and*" and "*or*" are underlined.

4.2 Syntactic States

Since the syntactic states are shared across all documents, we expect words associated with the syntactic states when applying HMM-LDA to the *Lectures* dataset to reflect the lecture style vocabulary.

In Figure 4, we list the top 10 words from each of the 19 syntactic states (state 20 is the topic state). Note that each state plays a clear syntactic role. For example, state 2 contains prepositions while state 7 contains verbs. Since the model is trained on transcriptions of spontaneous speech, hesitation disfluencies (<*uh*>, <*um*>, <*partial*>) are all grouped in state 3 along with other words (*so*, *if*, *okay*) that frequently indicate hesitation. While many of these hesitation words are conjunctions, the words in state 6 show that most conjunctions are actually assigned to a different state representing different syntactic behavior from hesitations. As demonstrated with spontaneous speech, HMM-LDA yields syntactic states that have a good correspondence to part-of-speech labels, without requiring any labeled training data.

4.3 Discussions

Although MCMC techniques converge to the global stationary distribution, we cannot guarantee convergence from observation of the perplexity alone. Unlike EM algorithms, random sampling may actually temporarily decrease the model likelihood. Thus, in the above analysis, the number of iterations was chosen to be at least double the point at which the perplexity first appeared to converge.

In addition to the number of iterations, the choice of the number of states and topics, as well as the values of the hyper-parameters on the Dirichlet prior, also impact the quality and effectiveness of the resulting model. Ideally, we run the algorithm with different combinations of the parameter values and perform model selection to choose the model with the best complexity-penalized likelihood. However, given finite computing resources, this approach is often im-

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
the	of	so	know	I	and	is	it's	it	a	way	it	two	going	that	can	very	to	have
this	in	<uh>	see	you	but	are	not	you	an	time	this	one	doing	what	will	more	just	be
a	for	if	do	we	or	was	that's	out	some	thing	that	three	one	how	would	little	longer	want
that	on	<um>	think	they	because	has	I'm	up	one	lot	there	hundred	looking	where	don't	much	doesn't	had
these	with	<partial>	go	let	as	were	just	them	no	question	which	m	sort	when	could	good	never	get
my	at	now	get	let's	that	goes	there's	that	in	kind	he	t	done	if	do	different	go	like
our	from	then	say	he	where	had	<uh>	me	two	point	here	five	able	why	just	than	physically	got
your	by	okay	make	I'll	thank	comes	we're	about	any	case	course	d	coming	which	me	important	that'll	need
those	about	well	look	people	which	means	also	here	this	idea	who	years	talking	as	should	long	anybody's	try
their	as	but	take	I'd	is	says	you're	all	another	problem	they	four	trying	because	may	as	with	take

Figure 4: The top 10 words from the 19 syntactic states computed from the *Lectures* dataset.

practical. As an alternative for future work, we would like to perform Gibbs sampling on the hyper-parameters (Griffiths et al., 2004) and apply the Dirichlet process to estimate the number of states and topics (Teh et al., 2004).

Despite the suboptimal choice of parameters and potential lack of convergence, the labels derived from HMM-LDA are still effective for language modeling applications, as described next.

5 Language Modeling Experiments

To evaluate the effectiveness of models derived from the separation of syntax from content, we performed experiments that compare the perplexities and WERs of various model combinations. For a baseline, we used an adapted model (L+T) that linearly interpolates trigram models trained on the *Lectures* (L) and *Textbook* (T) datasets. In all models, all interpolation weights and additional parameters are tuned on a development set consisting of the first half of the CS lectures and tested on the second half. Unless otherwise noted, modified Kneser-Ney discounting (Chen and Goodman, 1998) is applied with the respective training set vocabulary using the SRILM Toolkit (Stolcke, 2002).

To compute the word error rates associated with a specific language model, we used a speaker-independent speech recognizer (Glass, 2003). The lectures were pre-segmented into utterances by forced alignment of the reference transcription.

5.1 Lecture Style

In general, an n-gram model trained on a limited set of topic-specific documents tends to overemphasize words from the observed topics instead of evenly distributing weights over all potential topics. Specifically, given the list of words following an n-gram context, we would like to deemphasize the observed occurrences of topic words and ideally redistribute these counts to all potential topic words. As an approximation, we can build such a topic-deemphasized style trigram model (S) by using counts of only n-gram sequences that do not end on a topic word, smoothed over the *Lectures* vocabulary. Figure 5 shows the n-grams corresponding to an utterance used to build the style trigram model. Note that the counts of topic to style word transitions are not altered as these probabilities are mostly independent of the observed topic distribution.

By interpolating the style model (S) from above with the smoothed trigram model based on the *Lectures* dataset (L), the combined model (L+S) achieves a 3.6% perplexity reduction and 1.0% WER reduction over (L), as shown in Table 2. Without introducing topic-specific training data, we can already improve the generic lecture LM performance using the HMM-LDA labels.

<s> for the SPATIAL MEMORY </s>	
unigrams:	for, the, spatial, memory, </s>
bigrams:	<s> for, for the, ~~the spatial~~, ~~spatial memory~~, memory </s>
trigrams:	<s> <s> for, <s> for the, ~~for the spatial~~, ~~the spatial memory~~, spatial memory </s>

Figure 5: Style model n-grams. Topic words in the utterance are in uppercase.

5.2 Topic Domain

Unlike *Lectures*, the *Textbook* dataset contains content words relevant to the target lectures, but in a mismatched style. Commonly, the *Textbook* trigram model is interpolated with the generic model to improve the probability estimates of the transitions involving topic words. The interpolation weight is chosen to best fit the probabilities of these n-gram sequences while minimizing the mismatch in style. However, with only one parameter, all n-gram contexts must share the same mixture weight. Because transitions from contexts containing topic words are rarely observed in the off-topic *Lectures*, the *Textbook* model (T) should ideally have higher weight in these contexts than contexts that are more equally observed in both datasets.

One heuristic approach for adjusting the weight in these contexts is to build a topic-domain trigram model (D) from the *Textbook* n-gram counts with Witten-Bell smoothing (Chen and Goodman, 1998) where we emphasize the sequences containing a topic word in the context by doubling their counts. In effect, this reduces the smoothing on words following topic contexts with respect to lower-order models without significantly affecting the transitions from non-topic words. Figure 6 shows the adjusted counts for an utterance used to build the domain trigram model.

<s> HUFFMAN CODE can be represented as a BINARY TREE …	
unigrams:	huffman, code, can, be, represented, as, binary, tree, …
bigrams:	<s> huffman, huffman code (2×), code can (2×), can be, be represented, represented as, a binary, binary tree (2×), …
trigrams:	<s> <s> hufmann, <s> hufmann code (2×), hufmann code can (2×), code can be (2×), can be represented, be represented as, represented as a, as a binary, a binary tree (2×), ...

Figure 6: Domain model n-grams. Topic words in the utterance are in uppercase.

Empirically, interpolating the lectures, textbook, and style models with the domain model (L+T+S+D) further decreases the perplexity by 1.4% and WER by 0.3% over (L+T+S), validating our intuition. Overall, the addition of the style and domain models reduces perplexity and WER by a noticeable 7.1% and 2.1%, respectively, as shown in Table 2.

	Perplexity	
Model	**Development**	**Test**
L: Lectures Trigram	180.2 (0.0%)	199.6 (0.0%)
T: Textbook Trigram	291.7 (+61.8%)	331.7 (+66.2%)
S: Style Trigram	207.0 (+14.9%)	224.6 (+12.5%)
D: Domain Trigram	354.1 (+96.5%)	411.6 (+106.3%)
L+S	174.2 (–3.3%)	192.4 (–3.6%)
L+T: Baseline	138.3 (0.0%)	154.4 (0.0%)
L+T+S	131.0 (–5.3%)	145.6 (–5.7%)
L+T+S+D	128.8 (–6.9%)	143.6 (–7.1%)
L+T+S+D+Topic100		
• Static Mixture (cheat)	118.1 (–14.6%)	131.3 (–15.0%)
• **Dynamic Mixture**	**115.7 (–16.4%)**	**129.5 (–16.1%)**

	Word Error Rate	
Model	**Development**	**Test**
L: Lectures Trigram	49.5% (0.0%)	50.2% (0.0%)
L+S	49.2% (–0.7%)	49.7% (–1.0%)
L+T: Baseline	46.6% (0.0%)	46.7% (0.0%)
L+T+S	46.0% (–1.2%)	45.8% (–1.8%)
L+T+S+D	45.8% (–1.8%)	45.7% (–2.1%)
L+T+S+D+Topic100		
• Static Mixture (cheat)	45.5% (–2.4%)	45.4% (–2.8%)
• **Dynamic Mixture**	**45.4% (–2.6%)**	**45.6% (–2.4%)**

Table 2: Perplexity (top) and WER (bottom) performance of various model combinations. Relative reduction is shown in parentheses.

5.3 Textbook Topics

In addition to identifying content words, HMM-LDA also assigns words to a topic based on their distribution across documents. Thus, we can apply HMM-LDA with 100 topics to the *Textbook* dataset to identify representative words and their associated contexts for each topic. From these labels, we can build unsmoothed trigram language models (Topic100) for each topic from the counts of observed n-gram sequences that end in a word assigned to the respective topic.

Figure 7 shows a sample of the word n-grams identified via this approach for a few topics. Note that some of the n-grams are key phrases for the topic while others contain a mixture of syntactic and topic words. Unlike bag-of-words models that only identify the unigram distribution for each topic, the use of context-dependent labels enables the construction of n-gram topic models that not only characterize the frequencies of topic words, but also describe the transition contexts leading up to these words.

Huffman tree	Monte Carlo	time segment	assoc key
relative frequency	rand update	the agenda	the table
relative frequencies	random numbers	segment time	local table
the tree	trials remaining	current time	a table
one hundred	trials passed	first agenda	of records

Figure 7: Sample of n-grams from select topics.

5.4 Topic Mixtures

Since each target lecture generally only covers a subset of the available topics, it will be ideal to identify the specific topics corresponding to a target lecture and assign those topic models more weight in a linearly interpolated mixture model. As an ideal case, we performed a cheating experiment to measure the best performance of a statically interpolated topic mixture model (L+T+S+D+Topic100) where we tuned the mixture weights of all mixture components, including the lectures, textbook, style, domain, and the 100 individual topic trigram models on individual target lectures.

Table 2 shows that by weighting the component models appropriately, we can reduce the perplexity and WER by an additional 7.9% and 0.7%, respectively, over the (L+T+S+D) model even with simple linear interpolation for model combination.

To gain further insight into the topic mixture model, we examine the breakdown of the normalized topic weights for a specific lecture. As shown in Figure 8, of the 100 topic models, 15 of them account for over 90% of the total weight. Thus, lectures tend to show a significant topic skew which topic adaptation approaches can model effectively.

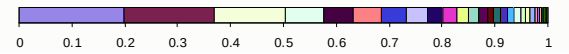

Figure 8: Topic mixture weight breakdown.

5.5 Topic Adaptation

Unfortunately, since different lectures cover different topics, we generally cannot tune the topic mixture weights ahead of time. One approach, without any a priori knowledge of the target lecture, is to adaptively estimate the optimal mixture weights as we process the lecture (Gildea and Hofmann, 1999). However, since the topic distribution shifts over a long lecture, modeling a lecture as an interpolation of components with fixed weights may not be the most optimal. Instead, we employ an exponential decay strategy where we update the current mixture distribution by linearly interpolating it with the posterior topic distribution given the current word. Specifically, applying Bayes' rule, the probability of topic *t* generating the current word *w* is given by:

$$P(t \mid w) = \frac{P(w|t)P(t)}{\sum_{t'} P(w|t')P(t')}$$

To achieve the exponential decay, we update the topic distribution after each word according to $P^{i+1}(t) = (1 - \alpha) \cdot P^{i}(t) + \alpha \cdot P(t \mid w^{i})$, where α is the adaptation rate.

We evaluated this approach of dynamic mixture weight adaptation on the (L+T+S+D+Topic 100) model, with the same set of components as the cheating experiment with static weights. As shown in Table 2, the dynamic model actually outperforms the static model by more than 1% in perplexity, by better modeling the dynamic topic substructure within the lecture.

To run the recognizer with a dynamic LM, we rescored the top 100 hypotheses generated with the (L+T+S+D) model using the dynamic LM. The WER obtained through such n-best rescoring yielded noticeable improvements over the (L+T+S+D) model without a priori knowledge of the topic distribution, but did not beat the optimal static model on the test set.

To further gain an intuition for mixture weight adaptation, we plotted the normalized adapted weights of the topic models across the first lecture of the test set in Figure 9. Note that the topic mixture varies greatly across the lecture. In this particular lecture, the lecturer starts out with a review of the previous lecture. Subsequently, he shows an example of computation using accumulators. Finally, he focuses the lecture on stream as a data structure, with an intervening example that finds pairs of *i* and *j* that sum up to a prime. By comparing the topic labels in Figure 9 with the top words from the corresponding topics in Figure 10, we observe that the topic weights obtained via dynamic adaptation match the subject matter of the lecture fairly closely.

Finally, to assess the effect that word error rate has on adaptation performance, we applied the adaptation algorithm to the corresponding transcript from the automatic speech recognizer (ASR). Traditional cache language models tend to be vulnerable to recognition errors since incorrect words in the history negatively bias the prediction of the current word. However, by adapting at a topic level, which reduces the number of dynamic parameters, the dynamic topic model is less sensitive to recognition errors. As seen in Figure 9, even with a word error rate around 40%, the normalized topic mixture weights from the ASR transcript still show a strong resemblance to the original weights from the manual reference transcript.

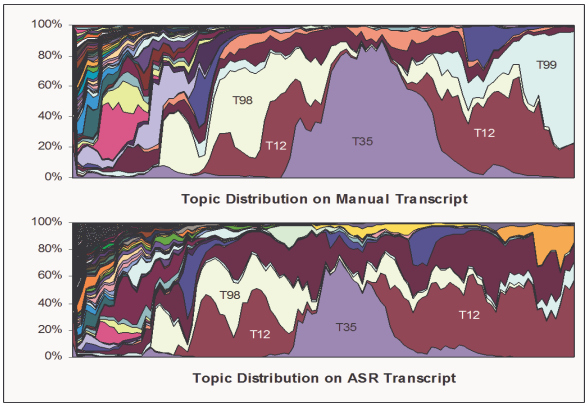

Figure 9: Adaptation of topic model weights on manual and ASR transcription of a single lecture.

T12	T35	T98	T99
stream	pairs	sequence	of
s	i	enumerate	see
streams	j	accumulate	and
integers	k	map	in
series	pair	interval	for
prime	s	filter	vs
filter	integers	sequences	register
delayed	sum	operations	data
interleave	queens	odd	as
infinite	t	nil	make

Figure 10: Top 10 words from select *Textbook* topics appearing in Figure 9.

6 Summary and Conclusions

In this paper, we have shown how to leverage context-dependent state and topic labels, such as the ones generated by the HMM-LDA model, to construct better language models for lecture transcription and extend topic models beyond traditional unigrams. Although the WER of the top recognizer hypotheses exceeds 45%, by dynamically updating the mixture weights to model the topic substructure within individual lectures, we are able to reduce the test set perplexity and WER by over 16% and 2.4%, respectively, relative to the combined *Lectures* and *Textbook* (L+T) baseline.

Although we primarily focused on lecture transcription in this work, the techniques extend to language modeling scenarios where exactly matched training data are often limited or nonexistent. Instead, we have to rely on appropriate combination of models derived from partially matched data. HMM-LDA and related techniques show great promise for finding structure in unlabeled data, from which we can build more sophisticated models.

The experiments in this paper combine models primarily through simple linear interpolation. As motivated in section 5.2, allowing for context-dependent interpolation weights based on topic

labels may yield significant improvement for both perplexity and WER. Thus, in future work, we would like to study algorithms for automatically learning appropriate context-dependent interpolation weights. Furthermore, we hope to improve the convergence properties of the dynamic adaptation scheme at the start of lectures and across topic transitions. Lastly, we would like to extend the LDA framework to support speaker-specific adaptation and apply the resulting topic distributions to lecture segmentation.

Acknowledgements

We would like to thank the anonymous reviewers for their useful comments and feedback. Support for this research was provided in part by the National Science Foundation under grant #IIS-0415865. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

Reference

Y. Akita and T. Kawahara. 2004. Language Model Adaptation Based on PLSA of Topics and Speakers. In *Proc. ICSLP*.

J. Bellegarda. 2000. Exploiting Latent Semantic Information in Statistical Language Modeling. In *Proc. IEEE*, 88(8):1279-1296.

D. Blei, A. Ng, and M. Jordan. 1993. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993-1022.

W. Buntine and A. Jakulin. 2005. Discrete Principal Component Analysis. Technical Report, Helsinki Institute for Information Technology.

S. Chen and J. Goodman. 1996. An Empirical Study of Smoothing Techniques for Language Modeling. In *Proc. ACL*, 310-318.

P. Clarkson and A. Robinson. 1997. Language Model Adaptation Using Mixtures and an Exponentially Decaying Cache. In *Proc. ICASSP*.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391-407.

S. Furui. 2003. Recent Advances in Spontaneous Speech Recognition and Understanding. In *Proc. IEEE Workshop on Spontaneous Speech Proc. and Rec*, 1-6.

D. Gildea and T. Hofmann. 1999. Topic-Based Language Models Using EM. In *Proc. Eurospeech.*

J. Glass. 2003. A Probabilistic Framework for Segment-based Speech Recognition. *Computer, Speech and Language*, 17:137-152.

J. Glass, T.J. Hazen, L. Hetherington, and C. Wang. 2004. Analysis and Processing of Lecture Audio Data: Preliminary Investigations. In *Proc. HLT-NAACL Workshop on Interdisciplinary Approaches to Speech Indexing and Retrieval*, 9-12.

J. Goodman. 2001. A Bit of Progress in Language Modeling (Extended Version). Technical Report, Microsoft Research.

T. Griffiths and M. Steyvers. 2004. Finding Scientific Topics. In *Proc. National Academy of Science*, 101(Suppl. 1):5228-5235.

T. Griffiths, M. Steyvers, D. Blei, and J. Tenenbaum. 2004. Integrating Topics and Syntax. *Adv. in Neural Information Processing Systems*, 17:537-544.

R. Iyer and M. Ostendorf. 1999. Modeling Long Distance Dependence in Language: Topic Mixtures Versus Dynamic Cache. In *IEEE Transactions on Speech and Audio Processing*, 7:30-39.

R. Kuhn and R. De Mori. 1990. A Cache-Based Natural Language Model for Speech Recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:570-583.

R. Lau, R. Rosenfeld, S. Roukos. 1993. Trigger-Based Language Models: a Maximum Entropy Approach. In *Proc. ICASSP*.

E. Leeuwis, M. Federico, and M. Cettolo. 2003. Language Modeling and Transcription of the TED Corpus Lectures. In *Proc. ICASSP*.

H. Nanjo and T. Kawahara. 2002. Unsupervised Language Model Adaptation for Lecture Speech Recognition. In *Proc. ICSLP*.

H. Nanjo and T. Kawahara. 2004. Language Model and Speaking Rate Adaptation for Spontaneous Presentation Speech Recognition. In *IEEE Trans. SAP*, 12(4):391-400.

A. Park, T. Hazen, and J. Glass. 2005. Automatic Processing of Audio Lectures for Information Retrieval: Vocabulary Selection and Language Modeling. In *Proc. ICASSP*.

M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. 2004. The Author-Topic Model for Authors and Documents. *20th Conference on Uncertainty in Artificial Intelligence*.

R. Rosenfeld. 1996. A Maximum Entropy Approach to Adaptive Statistical Language Modeling. *Computer, Speech and Language*, 10:187-228.

A. Stolcke. 2002. SRILM – An Extensible Language Modeling Toolkit. In *Proc. ICSLP*.

Y. Teh, M. Jordan, M. Beal, and D. Blei. 2006. Hierarchical Dirichlet Processes. To appear in *Journal of the American Statistical Association*.