Dynamic Nonlocal Language Modeling via Hierarchical Topic-Based Adaptation

Radu Florian and **David Yarowsky**
Computer Science Department and Center for Language and Speech Processing,
Johns Hopkins University
Baltimore, Maryland 21218
{rflorian,yarowsky}@cs.jhu.edu

Abstract

This paper presents a novel method of generating and applying hierarchical, dynamic topic-based language models. It proposes and evaluates new cluster generation, hierarchical smoothing and adaptive topic-probability estimation techniques. These combined models help capture long-distance lexical dependencies. Experiments on the Broadcast News corpus show significant improvement in perplexity (10.5% overall and 33.5% on target vocabulary).

1 Introduction

Statistical language models are core components of speech recognizers, optical character recognizers and even some machine translation systems Brown et al. (1990). The most common language modeling paradigm used today is based on *n-grams*, local word sequences. These models make a Markovian assumption on word dependencies; usually that word predictions depend on at most m previous words. Therefore they offer the following approximation for the computation of a word sequence probability: $P\left(w_1^N\right) = \prod_{i=1}^N P\left(w_i|w_1^{i-1}\right) \approx \prod_{i=1}^N P\left(w_i|w_{i-m+1}^{i-1}\right)$ where w_i^j denotes the sequence $w_i \ldots w_j$; a common size for m is 3 (trigram language models).

Even if n-grams were proved to be very powerful and robust in various tasks involving language models, they have a certain handicap: because of the Markov assumption, the dependency is limited to very short local context. Cache language models (Kuhn and de Mori (1992),Rosenfeld (1994)) try to overcome this limitation by boosting the probability of the words already seen in the history; trigger models (Lau et al. (1993)), even more general, try to capture the interrelationships between words. Models based on syntactic structure (Chelba and Jelinek (1998), Wright et al. (1993)) effectively estimate intra-sentence syntactic word dependencies.

The approach we present here is based on the observation that certain words tend to have different probability distributions in different topics. We propose to compute the conditional language model probability as a dynamic mixture model of K topic-specific language models:

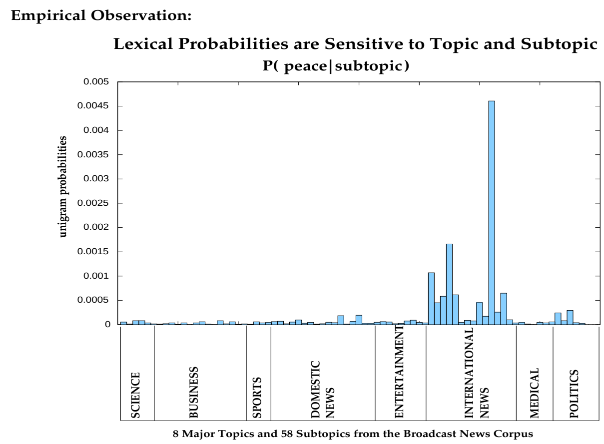


Figure 1: Conditional probability of the word *peace* given manually assigned Broadcast News topics

$$\begin{aligned} P\left(w_i|w_1^{i-1}\right) &= \sum_{t=1}^K P\left(t|w_1^{i-1}\right) \cdot P\left(w_i|t, w_1^{i-1}\right) \\ &\approx \sum_{t=1}^K P\left(t|w_1^{i-1}\right) \cdot P_t\left(w_i|w_{i-m+1}^{i-1}\right) \quad (1) \end{aligned}$$

The motivation for developing topic-sensitive language models is twofold. First, empirically speaking, many n-gram probabilities vary substantially when conditioned on topic (such as in the case of content words following several function words). A more important benefit, however, is that even when a given bigram or trigram probability is not topic sensitive, as in the case of sparse n-gram statistics, the topic-sensitive unigram or bigram probabilities may constitute a more informative backoff estimate than the single global unigram or bigram estimates. Discussion of these important smoothing issues is given in Section 4.

Finally, we observe that lexical probability distributions vary not only with topic but with subtopic too, in a hierarchical manner. For example, consider the variation of the probability of the word *peace* given major news topic distinctions (e.g. BUSINESS and INTERNATIONAL news) as illustrated in Figure 1. There is substantial subtopic probability variation for *peace* within INTERNATIONAL news (the word usage is 50-times more likely in INTERNATIONAL:MIDDLE-EAST than INTERNA-

TIONAL:JAPAN). We propose methods of hierarchical smoothing of $P(w_i|\text{topic}_t)$ in a topic-tree to capture this subtopic variation robustly.

1.1 Related Work

Recently, the speech community has begun to address the issue of topic in language modeling. Lowe (1995) utilized the hand-assigned topic labels for the Switchboard speech corpus to develop topic-specific language models for each of the 42 switchboard topics, and used a single topic-dependent language model to rescore the lists of N-best hypotheses. Error-rate improvement over the baseline language model of 0.44% was reported.

Iyer et al. (1994) used bottom-up clustering techniques on discourse contexts, performing sentence-level model interpolation with weights updated dynamically through an EM-like procedure. Evaluation on the Wall Street Journal (WSJ0) corpus showed a 4% perplexity reduction and 7% word error rate reduction. In Iyer and Ostendorf (1996), the model was improved by model probability reestimation and interpolation with a cache model, resulting in better dynamic adaptation and an overall 22%/3% perplexity/error rate reduction due to both components.

Seymore and Rosenfeld (1997) reported significant improvements when using a topic detector to build specialized language models on the Broadcast News (BN) corpus. They used TF-IDF and Naive Bayes classifiers to detect the most similar topics to a given article and then built a specialized language model to rescore the N-best lists corresponding to the article (yielding an overall 15% perplexity reduction using document-specific parameter re-estimation, and no significant word error rate reduction). Seymore et al. (1998) split the vocabulary into 3 sets: general words, on-topic words and off-topic words, and then use a non-linear interpolation to compute the language model. This yielded an 8% perplexity reduction and 1% relative word error rate reduction.

In collaborative work, Mangu (1997) investigated the benefits of using existing an Broadcast News topic hierarchy extracted from topic labels as a basis for language model computation. Manual tree construction and hierarchical interpolation yielded a 16% perplexity reduction over a baseline unigram model. In a concurrent collaborative effort, Khudanpur and Wu (1999) implemented clustering and topic-detection techniques similar on those presented here and computed a maximum entropy topic sensitive language model for the Switchboard corpus, yielding 8% perplexity reduction and 1.8% word error rate reduction relative to a baseline maximum entropy trigram model.

2 The Data

The data used in this research is the Broadcast News (BN94) corpus, consisting of radio and TV news transcripts form the year 1994. From the total of 30226 documents, 20226 were used for training and the other 10000 were used as test and held-out data. The vocabulary size is approximately 120k words.

3 Optimizing Document Clustering for Language Modeling

For the purpose of language modeling, the topic labels assigned to a document or segment of a document can be obtained either manually (by topic-tagging the documents) or automatically, by using an unsupervised algorithm to group similar documents in topic-like clusters. We have utilized the latter approach, for its generality and extensibility, and because there is no reason to believe that the manually assigned topics are optimal for language modeling.

3.1 Tree Generation

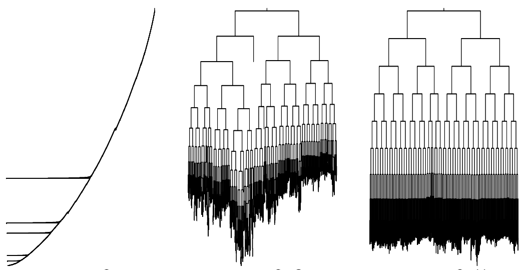
In this study, we have investigated a range of hierarchical clustering techniques, examining extensions of hierarchical agglomerative clustering, k-means clustering and top-down EM-based clustering. The latter underperformed on evaluations in Florian (1998) and is not reported here.

A generic hierarchical agglomerative clustering algorithm proceeds as follows: initially each document has its own cluster. Repeatedly, the two closest clusters are merged and replaced by their union, until there is only one top-level cluster. Pairwise document similarity may be based on a range of functions, but to facilitate comparative analysis we have utilized standard cosine similarity ($d(D_1, D_2) = \frac{\langle D_1, D_2 \rangle}{\|D_1\|_2 \|D_2\|_2}$) and IR-style term vectors (see Salton and McGill (1983)).

This procedure outputs a tree in which documents on similar topics (indicated by similar term content) tend to be clustered together. The difference between average-linkage and maximum-linkage algorithms manifests in the way the similarity between clusters is computed (see Duda and Hart (1973)). A problem that appears when using hierarchical clustering is that small centroids tend to cluster with bigger centroids instead of other small centroids, often resulting in highly skewed trees such as shown in Figure 2, α=0. To overcome the problem, we devised two alternative approaches for computing the intercluster similarity:

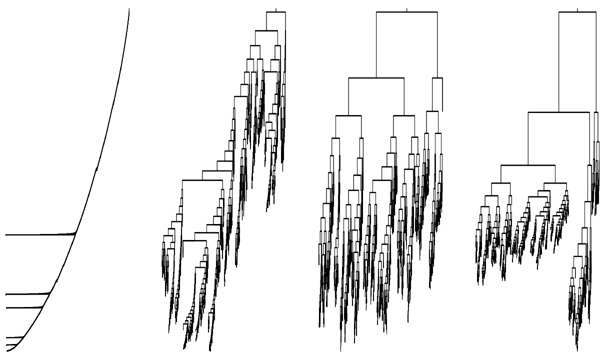
- Our first solution minimizes the attraction of large clusters by introducing a normalizing factor α to the inter-cluster distance function:

$$d(C_1, C_2) = \frac{< \mathrm{c}(C_1), \mathrm{c}(C_2) >}{N(C_1)^\alpha \, \|\mathrm{c}(C_1)\| \, N(C_2)^\alpha \, \|\mathrm{c}(C_2)\|} \quad (2)$$



$\alpha = 0$ $\alpha = 0.3$ $\alpha = 0.5$

Figure 2: As α increases, the trees become more balanced, at the expense of forced clustering



$\epsilon = 0$ $\epsilon = 0.15$ $\epsilon = 0.3$ $\epsilon = 0.7$

Figure 3: Tree-balance is also sensitive to the smoothing parameter ϵ.

where $N(C_k)$ is the number of vectors (documents) in cluster C_k and $c(C_i)$ is the centroid of the i^{th} cluster. Increasing α improves tree balance as shown in Figure 2, but as α becomes large the forced balancing degrades cluster quality.

- A second approach we explored is to perform basic smoothing of term vector weights, replacing all 0's with a small value ϵ. By decreasing initial vector orthogonality, this approach facilitates attraction to small centroids, and leads to more balanced clusters as shown in Figure 3.

Instead of stopping the process when the desired number of clusters is obtained, we generate the full tree for two reasons: (1) the full hierarchical structure is exploited in our language models and (2) once the tree structure is generated, the objective function we used to partition the tree differs from that used when building the tree. Since the clustering procedure turns out to be rather expensive for large datasets (both in terms of time and memory), only 10000 documents were used for generating the initial hierarchical structure.

[0]Section 3.2 describes the choice of optimum α.

3.2 Optimizing the Hierarchical Structure

To be able to compute accurate language models, one has to have sufficient data for the relative frequency estimates to be reliable. Usually, even with enough data, a smoothing scheme is employed to insure that $P\left(w_i|w_1^{i-1}\right) > 0$ for any given word sequence w_1^i.

The trees obtained from the previous step have documents in the leaves, therefore not enough word mass for proper probability estimation. But, on the path from a leaf to the root, the internal nodes grow in mass, ending with the root where the counts from the entire corpus are stored. Since our intention is to use the full tree structure to interpolate between the in-node language models, we proceeded to identify a subset of internal nodes of the tree, which contain sufficient data for language model estimation. The criteria of choosing the nodes for collapsing involves a goodness function, such that the *cut*[1] is a solution to a constrained optimization problem, given the constraint that the resulting tree has exactly k leaves. Let this evaluation function be $g(n)$, where n is a node of the tree, and suppose that we want to minimize it. Let $g(n,k)$ be the minimum cost of creating k leaves in the subtree of root n. When the evaluation function $g(n)$ satisfies the locality condition that it depends solely on the values $g(n_j,\cdot)$, (where $(n_j)_{j=1..k}$are the children of node n), $g(root)$ can be computed efficiently using dynamic programming[2] :

$$\begin{aligned} g(n,1) &= g(n) \\ g(n,k) &= \min_{\substack{j_1,,j_k \geq 1 \\ \sum_k j_k = k}} h\left(g\left(n_1,j_1\right),\ldots,g\left(n_k,j_k\right)\right) \end{aligned} \quad (3)$$

Let us assume for a moment that we are interested in computing a unigram topic-mixture language model. If the topic-conditional distributions have high entropy (e.g. the histogram of $P(w|topic)$ is fairly uniform), topic-sensitive language model interpolation will not yield any improvement, no matter how well the topic detection procedure works. Therefore, we are interested in clustering documents in such a way that the topic-conditional distribution $P(w|topic)$ is maximally skewed. With this in mind, we selected the evaluation function to be the *conditional entropy* of a set of words (possibly the whole vocabulary) given the particular classification. The conditional entropy of some set of words $\mathcal{W}$ given a partition $\mathcal{C}$ is

$$\begin{aligned} H(\mathcal{W}|\mathcal{C}) &= \sum_{i=1}^{n} P(C_i) \sum_{w \in \mathcal{W} \cap C_i} P(w|C_i) \cdot \log(P(w|C_i)) \\ &= \frac{1}{T} \sum_{i=1}^{n} \sum_{w \in \mathcal{W} \cap C_i} c(w,C_i) \cdot \log(P(w|C_i)) \end{aligned} \quad (4)$$

[1]the collection of nodes that collapse

[2]h is an operator through which the values $g(n_1,j_1),\ldots,g(n_k,j_k)$ are combined, as $\sum$ or $\prod$

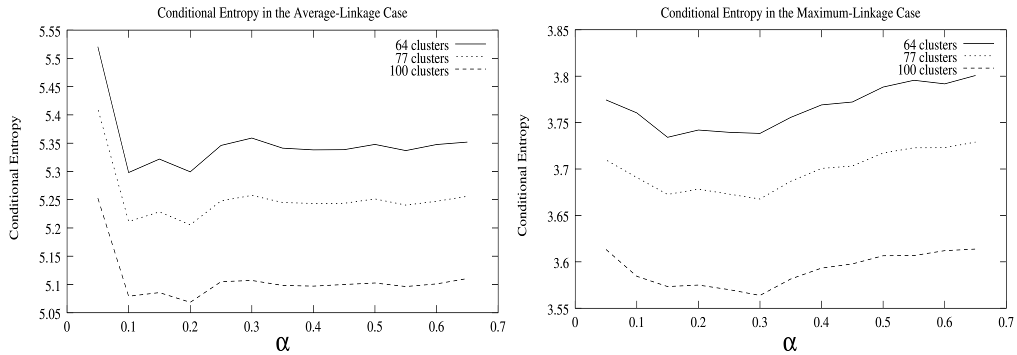


Figure 4: Conditional entropy for different α, cluster sizes and linkage methods

where $c(w, C_i)$ is the TF-IDF factor of word w in class C_i and T is the size of the corpus. Let us observe that the conditional entropy does satisfy the locality condition mentioned earlier.

Given this objective function, we identified the optimal tree cut using the dynamic-programming technique described above. We also optimized different parameters (such as α and choice of linkage method).

Figure 4 illustrates that for a range of cluster sizes, maximal linkage clustering with α=0.15-0.3 yields optimal performance given the objective function in equation (2).

The effect of varying α is also shown graphically in Figure 5. Successful tree construction for language modeling purposes will minimize the conditional entropy of $P(\mathcal{W}|\mathcal{C})$. This is most clearly illustrated for the word *politics*, where the tree generated with $\alpha = 0.3$ maximally focuses documents on this topic into a single cluster. The other words shown also exhibit this desirable highly skewed distribution of $P(\mathcal{W}|\mathcal{C})$ in the cluster tree generated when $\alpha = 0.3$.

Another investigated approach was k-means clustering (see Duda and Hart (1973)) as a robust and proven alternative to hierarchical clustering. Its application, with both our automatically derived clusters and Mangu's manually derived clusters (Mangu (1997)) used as initial partitions, actually yielded a small increase in conditional entropy and was not pursued further.

4 Language Model Construction and Evaluation

Estimating the language model probabilities is a two-phase process. First, the topic-sensitive language model probabilities $P\left(w_i|t, w_{i-m+1}^{i-1}\right)$ are computed during the training phase. Then, at run-time, or in the testing phase, topic is dynamically identified by computing the probabilities $P\left(t|w_1^{i-1}\right)$ as in section 4.2 and the final language model probabilities are computed using Equation (1). The tree used in the following experiments was generated using average-linkage agglomerative clustering, using parameters that optimize the objective function in Section 3.

4.1 Language Model Construction

The topic-specific language model probabilities are computed in a four phase process:

1. Each document is assigned to one leaf in the tree, based on the similarity to the leaves' centroids (using the cosine similarity). The document counts are added to the selected leaf's count.
2. The leaf counts are propagated up the tree such that, in the end, the counts of every internal node are equal to the sum of its children's counts. At this stage, each node of the tree has an attached language model - the relative frequencies.
3. In the root of the tree, a discounted Good-Turing language model is computed (see Katz (1987), Chen and Goodman (1998)).
4. m-gram smooth language models are computed for each node n different than the root by three-way interpolating between the m-gram language model in the parent $parent(n)$, the $(m-1)$-gram smooth language model in node n and the m-gram relative frequency estimate in node n:

$$\begin{aligned} &\hat{P}_n\left(w_m|w_1^{m-1}\right) = \\ &\lambda_n^1\left(w_1^{m-1}\right)\hat{P}_{\text{parent}(n)}\left(w_m|w_1^{m-1}\right) \\ &\quad+\lambda_n^2\left(w_1^{m-1}\right)\hat{P}_n\left(w_m|w_2^{m-1}\right) \\ &\quad+\lambda_n^3\left(w_1^{m-1}\right)f_n\left(w_m|w_1^{m-1}\right) \end{aligned} \tag{5}$$

with $\lambda_n^1\left(w_1^{m-1}\right) + \lambda_n^2\left(w_1^{m-1}\right) + \lambda_n^3\left(w_1^{m-1}\right) = 1$ for each node n in the tree. Based on how $\lambda_n^k\left(w_1^{m-1}\right)$ depend on the particular node n and the word history w_1^{m-1}, various models can be obtained. We investigated two approaches: a bigram model in which the λ's are fixed over the tree, and a more general trigram model in which $\lambda's$ adapt using an EM reestimation procedure.

- Case 1: $\mathrm{f}_{\mathrm{node}}(w_1) \neq 0$

$$\hat{P}_{\mathrm{node}}(w_2|w_1) = \begin{cases} P_{\mathrm{root}}(w_2|w_1) & \text{if } w_2 \in \mathcal{F}(w_1) \\ \lambda_1 \mathrm{f}_{\mathrm{node}}(w_2|w_1) \cdot \gamma_{\mathrm{node}}(w_1) + \lambda_2 \hat{P}_{\mathrm{node}}(w_2) & \\ \qquad + (1 - \lambda_1 - \lambda_2)\hat{P}_{\mathrm{parent(node)}}(w_2|w_1) & \text{if } w_2 \in \mathcal{R}(w_1) \\ \alpha_{\mathrm{node}}(w_1)\hat{P}_{\mathrm{node}}(w_2) & \text{if } w_2 \in \mathcal{U}(w1) \end{cases}$$

where

$$\gamma_{\mathrm{node}}(w_1) = \frac{1 - \sum_{w_2 \in \mathcal{F}(w_1)} \mathrm{f}_{\mathrm{node}}(w_2|w_1)}{(1+\beta) \sum_{w_2 \in \mathcal{R}(w_1)} \mathrm{f}_{\mathrm{node}}(w_2|w_1)}, \quad \alpha_{\mathrm{node}}(w_1) = \frac{\beta \left(1 - \sum_{w_2 \in \mathcal{F}(w_1)} \mathrm{f}_{\mathrm{node}}(w_2|w_1)\right)}{(1+\beta)\left(1 - \sum_{w_2 \in \mathcal{F}(w_1) \cup \mathcal{R}(w_1)} \hat{P}_{\mathrm{node}}(w_2)\right)}$$

- Case 2: $\mathrm{f}_{\mathrm{node}}(w_1) = 0$

$$\hat{P}_{\mathrm{node}}(w_2|w_1) = \begin{cases} P_{\mathrm{root}}(w_2|w_1) & \text{if } w_2 \in \mathcal{F}(w_1) \\ \lambda_2 \hat{P}_{\mathrm{node}}(w_2) \cdot \gamma_{\mathrm{node}}(w_1) & \\ \quad + (1 - \lambda_3)\hat{P}_{\mathrm{parent(node)}}(w_2|w_1) & \text{if } w_2 \in \mathcal{R}(w_1) \\ \alpha_{\mathrm{node}}(w_1)\hat{P}_{\mathrm{node}}(w_2) & \text{if } w_2 \in \mathcal{U}(w1) \end{cases}$$

where $\gamma_{\mathrm{node}}(w_1)$ and $\alpha_{\mathrm{node}}(w_1)$ are computed in a similar fashion such that the probabilities do sum to 1.

Figure 5: Basic Bigram Language Model Specifications

4.1.1 Bigram Language Model

Not all words are topic sensitive. Mangu (1997) observed that closed-class function words (FW), such as *the*, *of*, and *with*, have minimal probability variation across different topic parameterizations, while most open-class content words (CW) exhibit substantial topic variation. This leads us to divide the possible word pairs in two classes (topic-sensitive and not) and compute the λ's in Equation (5) in such a way that the probabilities in the former set are constant in all the models. To formalize this:

- $\mathcal{F}(w_1) = \{w_2 \in \mathcal{V} | (w_1, w_2)$ is fixed$\}$-the "fixed" space;
- $\mathcal{R}(w_1) = \{w_2 \in \mathcal{V} | (w_1, w_2)$ is free/variable$\}$-the "free" space;
- $\mathcal{U}(w_1) = \{w_2 \in \mathcal{V} | (w_1, w_2)$ was never seen$\}$-the "unknown" space.

The imposed restriction is, then: for every word $w_1$and any word $w_2 \in \mathcal{F}(w_1)$ $P_n(w_2|w_1) = P_{root}(w_2|w_1)$ in any node n.

The distribution of bigrams in the training data is as follows, with roughly 30% bigram probabilities allowed to vary in the topic-sensitive models:

Model	Bigram-type	Example	Freq.	
fixed	$p(FW\|FW)$	$p(the\|in)$	45.3%	least topic sensitive
fixed	$p(FW\|CW)$	$p(of\|scenario)$	24.8%	↓
free	$p(CW\|CW)$	$p(air\|cold)$	5.3%	↓
free	$p(CW\|FW)$	$p(air\|the)$	24.5%	most topic sensitive

This approach raises one interesting issue: the language model in the root assigns some probability mass to the unseen events, equal to the singletons' mass (see Good (1953),Katz (1987)). In our case, based on the assumptions made in the Good-Turing formulation, we considered that the ratio of the probability mass that goes to the unseen events and the one that goes to seen, free events should be fixed over the nodes of the tree. Let β be this ratio. Then the language model probabilities are computed as in Figure 5.

4.1.2 Ngram Language Model Smoothing

In general, n gram language model probabilities can be computed as in formula (5), where $\left(\lambda_n^k\left(w_1^{m-1}\right)\right)_{k=1\ldots3}$ are adapted both for the particular node n and history w_1^{m-1}. The proposed dependency on the history is realized through the history count $C\left(w_1^{m-1}\right)$ and the relevance of the history w_1^{m-1} to the topic in the nodes n and *parent* (n). The intuition is that if a history is as relevant in the current node as in the parent, then the estimates in the parent should be given more importance, since they are better estimated. On the other hand, if the history is much more relevant in the current node, then the estimates in the node should be trusted more. The mean adapted λ for a given height h is the tree is shown in Figure 6. This is consistent with the observation that splits in the middle of the tree tend to be most informative, while those closer to the leaves suffer from data fragmentation, and hence give relatively more weight to their parent. As before, since not all the m-grams are expected to be topic-sensitive, we use a method to insure that those m grams are kept "fixed" to minimize noise and modeling effort. In this case, though, 2 language models with different support are used: one that supports the topic insensitive m-grams and that is computed only once (it's a normalization of the topic-insensitive part of the overall model), and one

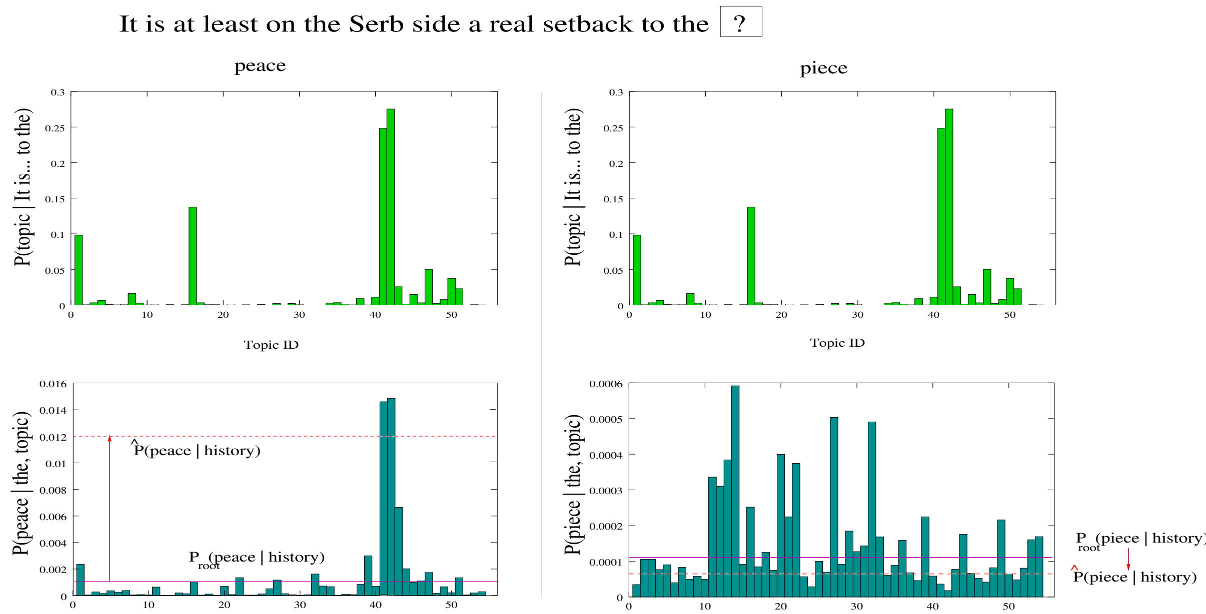


Figure 7: Topic sensitive probability estimation for **peace** and **piece** in context

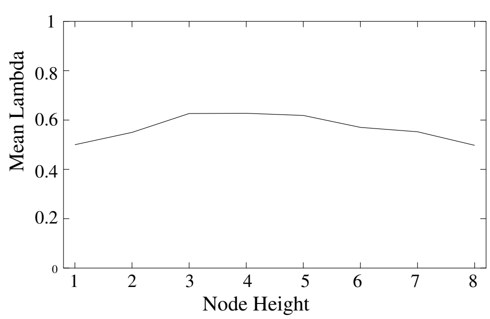


Figure 6: Mean of the estimated λs at node height h, in the unigram case

that supports the rest of the mass and which is computed by interpolation using formula (5). Finally, the final language model in each node is computed as a mixture of the two.

4.2 Dynamic Topic Adaptation

Consider the example of predicting the word following the Broadcast News fragment: "It is at least on the Serb side a real drawback to the $\boxed{\ ?\ }$". Our topic detection model, as further detailed later in this section, assigns a topic distribution to this left context (including the full previous discourse), illustrated in the upper portion of Figure 7. The model identifies that this particular context has greatest affinity with the empirically generated topic clusters #41 and #42 (which appear to have one of their foci on international events).

The lower portion of Figure 7 illustrates the topic-conditional bigram probabilities $P(w|\text{the}, \mathit{topic})$ for two candidate hypotheses for w: *peace* (the actually observed word in this case) and *piece* (an incorrect competing hypothesis). In the former case, $P(\mathit{peace}|\text{the}, \mathit{topic})$ is clearly highly elevated in the most probable topics for this context (#41,#42), and thus the application of our core model combination (Equation 1) yields a posterior joint product $P\left(w_i|w_1^{i-1}\right) = \sum_{t=1}^{K} P\left(t|w_1^{i-1}\right) \cdot P_t\left(w_i|w_{i-m+1}^{i-1}\right)$ that is 12-times more likely than the overall bigram probability, $P(\text{air}|\text{the}) = 0.001$. In contrast, the obvious accustically motivated alternative *piece,* has greatest probability in a far different and much more diffuse distribution of topics, yielding a joint model probability for this particular context that is 40% lower than its baseline bigram probability. This context-sensitive adaptation illustrates the efficacy of dynamic topic adaptation in increasing the model probability of the truth.

Clearly the process of computing the topic detector $P\left(t|w_1^{i-1}\right)$ is crucial. We have investigated several mechanisms for estimating this probability, the most promising is a class of normalized transformations of traditional cosine similarity between the document history vector w_1^{i-1} and the topic centroids:

$$P\left(t|w_1^{i-1}\right) = \frac{f\left(\text{Cosine-Sim}\left(t, w_1^{i-1}\right)\right)}{\sum_{t'} f\left(\text{Cosine-Sim}\left(t', w_1^{i-1}\right)\right)} \quad (6)$$

One obvious choice for the function f would be the identity. However, considering a linear contribution of similarities poses a problem: because topic detection is more accurate when the history is long, even unrelated topics will have a non-trivial contri-

Language Model						Perplexity on the entire vocabulary	Perplexity on the target vocabulary
Standard Bigram Model						**215**	**584**
Topic LMs	**History size**	**Scaled**	$g(x)$	$f(x)$	k-NN		
	100	yes	x	x^2	-	206	460
	1000	yes	x	x^2	-	195	405
	5000	yes*	x^*	x^{2*}	-*	**192** (-10%)	**389**(-33%)
	5000	yes	1	x	-	202	444
	5000	no	x	x^2	-	193	394
	5000	yes	x	x^2	15-NN	192	390
	5000	yes	e^x	xe^x	-	196	411

Table 1: Perplexity results for topic sensitive bigram language model, different history lengths

bution to the final probability[3], resulting in poorer estimates.

One class of transformations we investigated, that directly address the previous problem, adjusts the similarities such that closer topics weigh more and more distant ones weigh less. Therefore, f is chosen such that

$$\frac{f(x_1)}{f(x_2)} \leq \frac{x_1}{x_2} \text{ for } x_1 \leq x_2 \Leftrightarrow \\ \frac{f(x_1)}{x_1} \leq \frac{f(x_2)}{x_2} \text{ for } x_1 \leq x_2 \tag{7}$$

that is, $\frac{f(x)}{x}$ should be a monotonically increasing function on the interval $[0, 1]$, or, equivalently $f(x) = x \cdot g(x)$, g being an increasing function on $[0, 1]$. Choices for $g(x)$ include x, $x^\gamma (\gamma > 0)$, $log(x)$, e^x.

Another way of solving this problem is through the scaling operator $f'(x_i) = \frac{x_i - \min x_i}{\max x_i - \min x_i}$. By applying this operator, minimum values (corresponding to low-relevancy topics) do not receive any mass at all, and the mass is divided between the more relevant topics. For example, a combination of scaling and $g(x) = x^\gamma$ yields:

$$P\left(t_j | w_1^{i-1}\right) = \\ \frac{\left(\frac{Sim\left(w_1^{i-1}, t_j\right) - \min_k Sim\left(w_1^{i-1}, t_k\right)}{\max_k Sim\left(w_1^{i-1}, t_k\right) - \min_k Sim\left(w_1^{i-1}, t_k\right)}\right)^\gamma}{\sum_l \left(\frac{Sim\left(w_1^{i-1}, t_l\right) - \min_k Sim\left(w_1^{i-1}, t_k\right)}{\max_k Sim\left(w_1^{i-1}, t_k\right) - \min_k Sim\left(w_1^{i-1}, t_k\right)}\right)^\gamma} \tag{8}$$

A third class of transformations we investigated considers only the closest k topics in formula (6) and ignores the more distant topics.

4.3 Language Model Evaluation

Table 1 briefly summarizes a larger table of performance measured on the bigram implementation of this adaptive topic-based LM. For the default parameters (indicated by *), a statistically significant overall perplexity decrease of 10.5% was observed relative to a standard bigram model measured on the same 1000 test documents. Systematically modifying these parameters, we note that performance is decreased by using shorter discourse contexts (as histories never cross discourse boundaries, 5000-word histories essentially correspond to the full prior discourse). Keeping other parameters constant, $g(x) = x$ outperforms other candidate transformations $g(x) = 1$ and $g(x) = e^x$. Absence of k-nn and use of scaling both yield minor performance improvements.

It is important to note that for 66% of the vocabulary the topic-based LM is identical to the core bigram model. On the 34% of the data that falls in the model's target vocabulary, however, perplexity reduction is a much more substantial 33.5% improvement. The ability to isolate a well-defined target subtask and perform very well on it makes this work especially promising for use in model combination.

5 Conclusion

In this paper we described a novel method of generating and applying hierarchical, dynamic topic-based language models. Specifically, we have proposed and evaluated hierarchical cluster generation procedures that yield specially balanced and pruned trees directly optimized for language modeling purposes. We also present a novel hierarchical interpolation algorithm for generating a language model from these trees, specializing in the hierarchical topic-conditional probability estimation for a target topic-sensitive vocabulary (34% of the entire vocabulary). We also propose and evaluate a range of dynamic topic detection procedures based on several transformations of content-vector similarity measures. These dynamic estimations of $P(topic_i|history)$ are combined with the hierarchical estimation of $P(word_j|topic_i, history)$ in a product across topics, yielding a final probability estimate of $P(word_j|history)$ that effectively captures long-

[3]Due to unimportant word co-occurrences

distance lexical dependencies via these intermediate topic models. Statistically significant reductions in perplexity are obtained relative to a baseline model, both on the entire text (10.5%) and on the target vocabulary (33.5%). This large improvement on a readily isolatable subset of the data bodes well for further model combination.

Acknowledgements

The research reported here was sponsored by National Science Foundation Grant IRI-9618874. The authors would like to thank Eric Brill, Eugene Charniak, Ciprian Chelba, Fred Jelinek, Sanjeev Khudanpur, Lidia Mangu and Jun Wu for suggestions and feedback during the progress of this work, and Andreas Stolcke for use of his hierarchical clustering tools as a basis for some of the clustering software developed here.

References

P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, F. Jelinek, J. Lafferty, R. Mercer, and P. Roossin'. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2).

Ciprian Chelba and Fred Jelinek. 1998. Exploiting syntactic structure for language modeling. In *Proceedings COLING-ACL*, volume 1, pages 225–231, August.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techinques for language modeling. Technical Report TR-10-98, Center for Research in Computing Technology, Harvard University, Cambridge, Massachusettes, August.

Richard O. Duda and Peter E. Hart. 1973. *Patern Classification and Scene Analysis*. John Wiley & Sons.

Radu Florian. 1998. Exploiting nonlocal word relationships in language models. Technical report, Computer Science Department, Johns Hopkins University. http://nlp.cs.jhu.edu/~rflorian/papers/topic-lm-tech-rep.ps.

J. Good. 1953. The population of species and the estimation of population parameters. *Biometrica*, 40, parts 3,4:237–264.

Rukmini Iyer and Mari Ostendorf. 1996. Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models. In *Proceedings of the International Conferrence on Spoken Language Processing*, volume 1, pages 236–239.

Rukmini Iyer, Mari Ostendorf, and J. Robin Rohlicek. 1994. Language modeling with sentence-level mixtures. In *Proceedings ARPA Workshop on Human Language Technology*, pages 82–87.

Slava Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. In *IEEE Transactions on Acoustics, Speech, and Signal Processing, 1987*, volume ASSP-35 no 3, pages 400–401, March 1987.

Sanjeev Khudanpur and Jun Wu. 1999. A maximum entropy language model integrating n-gram and topic dependencies for conversational speech recognition. In *Proceedings of ICASSP*.

R. Kuhn and R. de Mori. 1992. A cache based natural language model for speech recognition. *IEEE Transaction PAMI*, 13:570–583.

R. Lau, Ronald Rosenfeld, and Salim Roukos. 1993. Trigger based language models: a maximum entropy approach. In *Proceedings of ICASSP*, pages 45–48, April.

S. Lowe. 1995. An attempt at improving recognition accuracy on switchboard by using topic identification. In *1995 Johns Hopkins Speech Workshop, Language Modeling Group, Final Report*.

Lidia Mangu. 1997. Hierarchical topic-sensitive language models for automatic speech recognition. Technical report, Computer Science Department, Johns Hopkins University. http://nlp.cs.jhu.edu/~lidia/papers/tech-rep1.ps.

Ronald Rosenfeld. 1994. A hybrid approach to adaptive statistical language modeling. In *Proceedings ARPA Workshop on Human Language Technology*, pages 76–87.

G. Salton and M. McGill. 1983. *An Introduction to Modern Information Retrieval*. New York, McGram-Hill.

Kristie Seymore and Ronald Rosenfeld. 1997. Using stopy topics for language model adaptation. In *EuroSpeech97*, volume 4, pages 1987–1990.

Kristie Seymore, Stanley Chen, and Ronald Rosenfeld. 1998. Nonlinear interpolation of topic models for language model adaptation. In *Proceedings of ICSLP98*.

J. H. Wright, G. J. F. Jones, and H. Lloyd-Thomas. 1993. A consolidated language model for speech recognition. In *Proceedings EuroSpeech*, volume 2, pages 977–980.