Large-scale Topic Detection And Language Model Adaptation

Kristie Seymore Ronald Rosenfeld

June 1997

CMU-CS-97-152

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This research was sponsored by the Department of the Navy, Naval Research Laboratory under Grant No. N00014-93-1-2005, the National Security Agency under Grant numbers MDA904-96-1-0113 and MDA904-97-1-0006, and under a National Science Foundation Graduate Research Fellowship.

Keywords: speech recognition, statistical language modeling, topic detection, topic adaptation, document clustering

Abstract

The subject matter of any conversation or document can typically be described as some combination of elemental topics. We have developed a language model adaptation scheme that takes a piece of text, chooses the most similar topic clusters from a set of over 5000 elemental topics, and uses topic specific language models built from the topic clusters to rescore N-best lists. We are able to achieve a 15% reduction in perplexity and a small improvement in word error rate by using this adaptation. We also investigate the use of a topic tree, where the amount of training data for a specific topic can be judiciously increased in cases where the elemental topic cluster has too few word tokens to build a reliably smoothed and representative language model. Our system is able to fine-tune topic adaptation by interpolating models chosen from thousands of topics, allowing for adaptation to unique, previously unseen combinations of subjects.

1 Introduction

In this paper, we explore large-scale, fine-tunable topic adaptation for statistical language modeling. We are interested in taking the initial transcription of a story supplied by a speech recognizer, identifying a set of topics that describe the content of the story by choosing topic-specific subsets of the language model training text, building a language model from each of the selected subsets, interpolating these models at the word level, and using the new language model score to reevaluate speech recognition hypotheses. The goal of the adaptation is to lower the word error rate (WER) of the story transcription output by the speech recognizer by providing language model scores that reflect a higher expectation of words and word-sequences that are characteristic of the identified topics of the story. This adaptation can be described as large-scale because the most similar topics to a new piece of text are chosen from a set of over 5000 topic candidates. One strength of this approach is the ability for diverse, typically unrelated topics to be selected and interpolated together to match the unique events present in a new story. Previously unseen combinations of topics occur frequently in domains such as Broadcast News, where current events dictate the contents of each article.

2 Topic Adaptation

The topic adaptation scheme we are using consists of the following steps:

1. Stories from an annotated corpus that share similar topics are gathered together into a set of clusters based on manually-assigned keywords.

2. A classifier is used to find the clusters that are most similar in topic to a story transcription output by a speech recognizer.

3. Language models are built from each of the clusters of data found to be the most similar to the new story.

4. The language models are interpolated at the word level and the interpolated score is used to rescore the speech recognizer's hypotheses in an N-best framework.

Each of these steps will be reviewed in detail in the following sections.

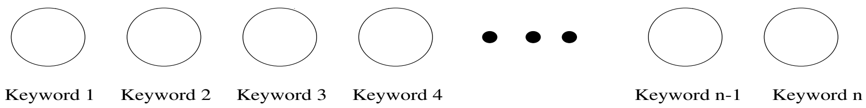

Figure 1: Keyword-based topic clusters.

2.1 Clustering

Given a corpus with story boundaries marked and manually-chosen keywords assigned to each story, topic clusters are created by defining each unique keyword as a label for a cluster, as in Figure 1. Each keyword represents an elemental topic, and all stories that have that keyword are assigned to its particular cluster. Each cluster is then a candidate to be used in topic adaptation.

Topic trees can be built from the topic clusters by treating the clusters as leaves and iteratively merging the topics together to form a tree, as in Figure 2. When two clusters are merged together, the resulting node in the tree has the benefit of more training data with which to estimate language model parameters, but is more general in topic than the children clusters. We can use the topic tree structure to combine the advantages of having larger clusters for parameter estimation and smaller clusters for topic focus. Each path from leaf to root specifies a set of nodes that start out in a very distinct topic and then gradually become more general as the clusters become larger. At runtime, automatic topic identification is performed on a decoded document and results in a small number of active leaf topics. Language models built at various nodes along the active paths can be combined to best model the current document. The construction of topic trees has been explored in the Switchboard domain by Carlson [1].

Agglomerative clustering has been used successfully for topic adaptation in a mixture modeling framework [2, 3]. In these cases, training data was partitioned into a relatively small set of topic clusters (less than one hundred.) One advantage of retaining thousands of individual topic clusters is the ability to make fine distinctions between different subjects and mix unusual topics together that may occur in a future story.

An important feature of creating topic clusters based on keywords is the presence of data overlap between clusters. If one story contains five different keywords describing its content, then the text for the story will appear in five different clusters. When using agglomerative clustering to create a topic tree, the effects of data overlap on the measure of cluster similarity need to be considered. In this

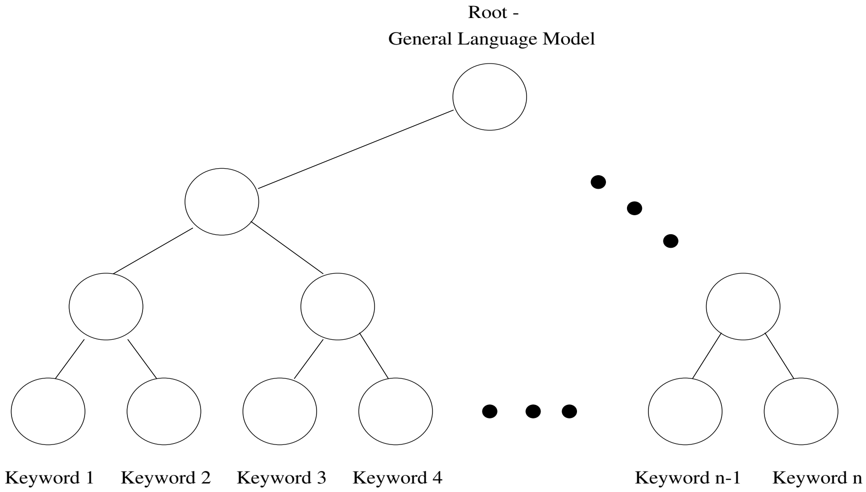

Figure 2: A topic tree built from keyword clusters.

work, no corrective action was taken to account for the similarity measure bias due to data overlap. Possible solutions include excluding the overlapping data from all similarity calculations, assigning half of each duplicated story to each leaf, or using supervised clustering to make reasonable decisions.

2.2 Topic Detection

Once we have a set of topic clusters, we can use topic detection to determine the most topic-similar clusters to a new piece of text. We consider two topic detection methods: the TFIDF classifier and the naive Bayes classifier. Both methods input a story and output the list of topic clusters ranked in order of decreasing similarity. Even when the text given to the classifiers contains word errors, as is the case when we use speech recognition hypotheses for detection, topic detection will still perform reasonably well, as we will show below. As long as the word errors in the hypothesis are not significantly topic-correlated, the correct content words in the hypothesis will provide enough evidence for the selection of appropriate clusters.

2.2.1 The TFIDF Classifier

The TFIDF measure [4] assigns a weight to each unique word in a document representing how topic-specific that word is to its document or cluster. If a cluster contains t distinct words, the cluster text can be represented as a t-dimensional vector of weights $\mathbf{D}_i = (w_{i1}, w_{i2}, w_{i3}, \ldots, w_{it})$, where each weight is given by:

$$w_{ik} = tf_{ik} \log(N/n_k) \tag{1}$$

The term frequency, tf_{ik} is the number of times that word k appears in cluster i. The inverse document frequency component computes the log of the ratio of N, the total number of clusters, to n_k, the number of clusters containing word k. This weighting function assigns high values to topic specific words, which are those words that appear with high frequency within one cluster but appear in relatively few other clusters. Words that occur in many clusters, or that occur with low frequency, are deemed more general and are assigned low weights.

Given some new text represented by weight vector $\mathbf{D}_j$, the topic similarity between cluster i and the new text can be computed with the following cosine measure:

$$sim(\mathbf{D}_i, \mathbf{D}_j) = \frac{\sum_{k=1}^{t} w_{ik} w_{jk}}{\sqrt{\sum_{k=1}^{t} (w_{ik})^2 \sum_{k=1}^{t} (w_{jk})^2}} \tag{2}$$

Equation 2 computes the cosine of the angle between the two vectors representing the two sets of text. It is normalized for vector length, so that large clusters are not favored. This similarity measure produces a high value when the two texts being compared are similar, with a value of 1 when they are identical. A similarity value of zero means that the topics of the texts are unrelated.

2.2.2 Naive Bayes Classifier

A naive Bayes classifier calculates the probability of a topic given the words in a new document. We make the traditional simplifying assumption that words in the document occur independently of one another in Equation 3.

$$p(topic \mid doc) \propto p(topic) \prod_{w_i \in doc} p_{smoothed}(w_i \mid topic) \tag{3}$$

The topic priors are computed from the topic document frequencies and the probability of a word given a topic is computed by smoothing the unigram distribution within the topic cluster with the general unigram distribution obtained from the entire training corpus, as shown in Equation 4. The smoothing parameter α is empirically chosen.

$$p_{smoothed}(w_i \mid topic) = (1-\alpha)p(w_i \mid topic) + \alpha p(w_i) \tag{4}$$

Many other topic detection techniques exist. Imai et al. have developed a Hidden Markov Model system for topic detection which identifies multiple topics per story and considers that each word in the story need not be related to all of the story's topics [5]. Joachims analyzes several topic detection algorithms, including TFIDF and the naive Bayes classifier, in [6].

2.3 Language Models

In the speech recognition paradigm, each time a new story is decoded an initial hypothesis transcription is produced. We then feed the hypothesis transcription to the classifier, which chooses the most similar topic clusters. Language models are built from the text in each of the selected clusters. Here, Good-Turing discounted trigram backoff models [7] using all bigrams and trigrams (no cutoffs) were built with the CMU Statistical Language Modeling toolkit [8].

2.4 Model Interpolation

The individual language models built from the chosen clusters (or from nodes farther up in the tree when a topic tree is being used) are interpolated together at the word level to produce a new language score, as in Equation 5.

$$\hat{p}_{new}(w_i \mid w_{i-2}, w_{i-1}) = \sum_{j=1}^{k} \lambda_j \hat{p}_j(w_i \mid w_{i-2}, w_{i-1}) \tag{5}$$

Here, k is the number of models being interpolated, λ_j is the interpolation weight for model j, and $\sum_{j=1}^{k} \lambda_j = 1$. The speech recognition hypotheses are then reevaluated in an N-best paradigm according to the new language scores.

3 Experiments

The training data used in these experiments is the Broadcast News corpus obtained from Primary Source Media [9]. The data used here covers the period from 1992 -

1995 and consists of 130 million words of news reports and interviews from ABC News, CNN, PBS, and National Public Radio. Story boundaries are marked, and each story is accompanied by a set of keywords (4 to 5 on average) that describe the story's content. The corpus was split into topic clusters by collecting the keywords from all stories and assigning each keyword to a cluster. The text for each story was assigned to the clusters of the story's keywords. Many of the keywords have sub-categories, in which case the sub-categories were separated from the main keyword and treated as keywords themselves. Summary stories, keywords with only one story and certain geographic keywords were excluded, resulting in 5883 topic clusters. A sample list of keywords is shown in Table 1.

Gems
General Agreement on Tariffs and Trade
General Dynamics Corp.
General Electric Co.
General Mills
General Motors Corp.
General Motors automobiles
Generals
Generation gap
Generic drugs
Generic products
Genetic counseling
Genetic engineering
Genetics
Genital mutilization
Genocide
Genovese, Kitty
Geography
Geology
George (Periodical)
George Washington University

Table 1: Sample of topic cluster keywords. Each keyword represents a topic cluster.

The most frequent 63k words from the four years of Broadcast News text defined the vocabulary for calculating cluster similarity. Twenty Broadcast News articles obtained from the Linguistic Data Consortium's (LDC) release of the Broadcast News corpus were randomly selected from the period covering January 1996 through April 1996 as a test set to compare the TFIDF and naive Bayes classifiers. Each of these twenty articles contained a minimum of 500 word tokens and at least one manually assigned keyword from the list of 5883 topic clusters. The development and evaluation sets from the 1996 ARPA Hub4 continuous speech recognition evaluation were used as speech recognition test sets. These sets contain story boundaries, where each boundary indicates a change in topic. The development set contains 57 stories and the evaluation set contains 74 stories. The number of word tokens in each story from the development and evaluation sets ranges from 6 to 2131.

3.1 Topic Detection Experiments

The TFIDF and naive Bayes classifiers were used for topic detection on the twenty-story test set from 1996. The Bayes classifier used $\alpha = 0.25$. Each classifier compared each test story to the 5883 topic leaf clusters and generated a ranked list of topic clusters in order of decreasing similarity to the test story. The "correct" topics for each test story were the manually assigned keywords that accompanied each story that were also found among the 5883 leaf clusters. Precision and recall results at 5, 10 and 20 were calculated as in [10] and are shown in Table 2. For this task, the naive Bayes classifier outperforms the TFIDF classifier across all three levels of precision and recall.

The largest story from the Hub4 development set consists of 2131 words and discusses suspicions of drug use by Chinese swimmers during the 1996 Olympics. The correct story transcript and the errorful first-pass Sphinx III [11] recognition hypotheses for this story (45% WER) were classified using both the TFIDF measure and the naive Bayes classifier. The 10 most similar clusters chosen by the TFIDF measure for the correct and errorful transcripts are shown in Table 3. The 10 most similar clusters chosen by the naive Bayes classifier for both transcripts are shown in Table 4.

Both classification methods choose reasonable topics when using either the correct or errorful story transcripts. For both classifiers, six of the clusters chosen when using the correct transcript are also chosen when using the errorful transcript. It is interesting to note that the two methods seem to choose slightly different

	TFIDF	Bayes
Precision at 5	49.0%	59.0%
Precision at 10	32.5%	42.0%
Precision at 20	22.5%	24.3%
Recall at 5	48.5%	56.1%
Recall at 10	62.8%	79.1%
Recall at 20	82.0%	89.2%

Table 2: Precision and Recall values at 5, 10 and 20 for the TFIDF and Naive Bayes classifiers.

types of clusters. In this case, the TFIDF classifier chooses many clusters about China, whereas the naive Bayes classifier chooses more sports-related clusters. Most importantly, we see that the clusters chosen by either method when using a transcript with a high word error rate are related to the topic of the story.

3.2 Perplexity Reduction

In order to determine the best way to interpolate topic specific language models, we varied the number of topic specific models chosen per story for adaptation and measured development set perplexity. First, topic detection was run using the TFIDF and naive Bayes classifiers on errorful first-pass Sphinx III recognition hypotheses from each of the 57 stories from the development set. The word error rate (WER) of the development set was 40%. A 51k vocabulary general trigram backoff language model was built from LDC's release of the Broadcast News corpus. Good-Turing discounted trigram backoff language models were built from each of the 20 most similar topic clusters chosen by the classifiers for each development set story. The perplexity for each story was computed by interpolating the most similar 5, 10 or 20 topic models for each story with the 51k general language model at the word level. Model interpolation weights were obtained with the EM algorithm and perplexity was computed using two-way cross validation. All of the story perplexities were combined (at the entropy level to adjust for different numbers of word tokens) to give a final development set perplexity. Results are shown in Table 5. Using twenty topic models chosen by the naive Bayes classifier yields the greatest reduction in perplexity over the

TFIDF Classifier	
Correct Transcript	Errorful Transcript (45% WER)
China	China
Olympic Games	Favored nation clause
Olympic Games, Barcelona, 1992	Chinese Americans
Favored nation clause	Olympic Games
Chinese Americans	Intellectual property rights
Drug testing	Chinese in the United States
Olympic Games, Atlanta, 1996	Olympic Games, Barcelona, 1992
Intellectual property rights	Wu, Harry
Swimming	Civil rights
Athletes	Zemin, Jiang

Table 3: Ten most similar clusters chosen with TFIDF, correct and errorful transcripts.

general Broadcast News model from 222 to 188, a 15% reduction.

Next, we built two topic trees. The first tree was built automatically by merging the 5883 topic leaf clusters iteratively to the root. At each iteration, the node with the fewest words was chosen to be merged with its most similar node, which was chosen by the TFIDF classifier. The second tree was built in the same way as the first, except that if the similarity value between the smallest cluster and its most similar cluster was below a threshold of 0.3, the smallest cluster was 'orphaned', or linked directly to the root. The orphan tree did not force a merge if no good match existed, whereas the automatic tree forced a merge at each iteration.

The 5883 leaf clusters range in token size from 393 to 6,234,183. Two hundred thirty of the 5883 leaf clusters contain less than one thousand word tokens. In cases where so few tokens are available, adaptation may benefit from using more data. In an effort to verify this hypothesis, three development set stories and one of the most similar leaves for each story were selected. For each of the three story-leaf pairs, language models were built at various nodes along the path from leaf to root for both the automatic tree and the orphan tree. Each model was interpolated with the 51k general model, and the perplexity of the story was computed using two-way cross-validation. In all cases, the perplexity decreased or stayed the same when a model built from a node with more data than the leaf cluster was used, as

Naive Bayes Classifier	
Correct Transcript	Errorful Transcript (45% WER)
Olympic Games, Barcelona, 1992	Olympic Games
Olympic Games	Olympic Games, Barcelona, 1992
Drug testing	China
Athletes	Athletics
Sports	Drug testing
Gymnastics	Olympic Games, Sydney, 2000
Louganis, Greg	Gymnastics
Athletics	Running races
Diving	Athletes
Olympic Games, Seoul, 1988	Wu, Harry

Table 4: Ten most similar clusters chosen with naive Bayes classifier, correct and errorful transcripts.

shown in Tables 6 and 7. For example, interpolating a leaf cluster language model built from 35,680 tokens with the general language model results in a perplexity of 219, whereas interpolating a language model built from a node located higher up the path with 100,500 tokens with the general language model results in a perplexity of 210. This limited example demonstrates that at least in some cases when interpolating only one leaf with the general language model per story, adding additional relevant text is helpful.

Topic tree adaptation was tested on the development set stories by setting token cutoffs. In all cases, twenty leaf clusters were considered per story. For both trees

General model		222
Leaves	TFIDF	Bayes
5	193	193
10	191	189
20	189	188

Table 5: Development set perplexity, leaves only.

Paths in automatic tree					
Tokens	PP	Tokens	PP	Tokens	PP
13445	233	266125	201	35680	219
25353	229	300170	200	100500	210
64820	225	451893	202	574818	226
100500	227			1002910	233
574818	223				
Root	264	Root	220	Root	272

Table 6: Perplexity variation moving up automatic tree paths from leaf to root.

Paths in orphan tree					
Tokens	PP	Tokens	PP	Tokens	PP
13445	233	266125	201	35680	219
25353	229	305562	201	96495	210
60815	225	333591	202		
96495	226				
Root	264	Root	220	Root	272

Table 7: Perplexity variation moving up orphan tree paths from leaf to root.

(automatic and orphan), whenever a leaf cluster was chosen for interpolation, the topic model was built from the lowest node in the path from leaf to root that had at least as many word tokens as the predetermined threshold. These nodes are referred to as 'active nodes' in the discussion below. Thresholds of 50k and 200k were set. Occasionally the paths for similar leaves merge, and in these cases less than twenty models were interpolated for those stories. The general broadcast news model (i.e. the model at the root of the tree) was always interpolated with the topic models.

In the case of the orphan tree, sometimes the node just below the root in an active path had fewer tokens than the threshold, leaving only the root node with enough tokens for interpolation. Therefore, two orphan tree scenarios were evaluated: in the first, all paths that assigned the root as the active node (because all other nodes in the path had fewer tokens than the threshold) were left out completely, meaning that the selected leaf did not contribute a model for interpolation. In the second scenario (designated by '+leaves'), all paths that assigned the root as the active node built the topic model from the leaf of the path, even though there were fewer tokens in the leaf node than the threshold. Perplexity results for these cases are shown in Tables 8 and 9. In all cases, interpolating topic models results in a decrease in perplexity over using only the general trigram model. Generally, none of the tree scenarios works as well as interpolating only the leaves, except for the Bayes orphan tree '+leaves' cases, which perform as well as the twenty Bayes leaves.

General model		222
Token thresh	TFIDF	Bayes
Leaves only	189	188
50k	191	189
200k	192	191

Table 8: Development set perplexity, automatic tree.

General model		222
Token thresh	TFIDF	Bayes
Leaves only	189	188
50k	191	189
50k+leaves	190	188
200k	196	192
200k+leaves	191	188

Table 9: Development set perplexity, orphan tree.

3.3 N-best Rescoring

Next, we wanted to see if using these models to rescore N-best lists would lead to a reduction in recognition WER. Two interpolation weighting schemes were tested. In the first, indicated by 'min PP', the cluster language models and the 51k general language model were interpolated with weights obtained by minimizing the perplexity of the errorful first-pass decoder hypothesis. The second interpolation scheme, 'uniform', assigned a weight of 0.55 to the general 51k language model and uniform interpolation weights to the remaining topic models. Rescoring consisted of using the original acoustic score, the new language model score, and a word insertion penalty. For the development set, N = 500, and the for the evaluation set, N = 200. Filled pauses were predicted from manually set unigram probabilities [12]. For the development set, the first-pass WER with no rescoring was 40.2%. The lowest N-best WER, found by using the reference transcripts to choose the N-best hypotheses with the lowest error, was 34.6%. The lowest N-best WER represents an upper bound on the performance of N-best rescoring. Using just the 51k general language model to rescore results in a WER of 40.1%. Language model score and insertion penalty weights were chosen by two-way cross validation, and the average weight values were used for evaluation set rescoring. The evaluation N-best lists were generated after two passes of the Sphinx III decoder. Topic adaptation scenarios tested with rescoring include twenty TFIDF-chosen leaves, twenty Bayes-chosen leaves and the Bayes orphan '+leaves' topic tree with a token threshold of 200k. Rescoring results are shown in Tables 10 and 11.

Condition	WER
No topic adaptation	40.2%
Lowest N-best WER	34.6%
General trigram	40.1%
TFIDF leaves, min PP	39.6%
TFIDF leaves, uniform	39.7%
Bayes leaves, min PP	39.5%
Bayes leaves, uniform	39.5%
Bayes orphan tree 200k+leaves, min PP	39.6%
Bayes orphan tree 200k+leaves, uniform	39.6%

Table 10: Development set word error rate using different language scores.

For both the evaluation and development sets, there is no large WER difference between using uniform model interpolation weights or choosing weights by

Condition	WER
2nd pass decoder output	35.5%
TFIDF leaves, min PP	35.3%
TFIDF leaves, uniform	35.5%
Bayes leaves, min PP	35.4%
Bayes leaves, uniform	35.4%
Bayes orphan tree 200k+leaves, min PP	35.3%
Bayes orphan tree 200k+leaves, uniform	35.5%

Table 11: Evaluation set word error rate.

minimizing perplexity. Rescoring the N-best lists with the topic score from the interpolation of Bayes-chosen leaves results in the greatest decrease in WER over the original 1st pass transcription (no adaptation) on the development set. In this case the error rate drops from 40.2% to 39.5%. Adaptation with either the TFIDF-chosen leaves or the orphan tree lowers the WER to 39.6%. However, none of the topic scores results in a significant improvement in WER on the evaluation set. Adaptation on the evaluation set with Bayes-chosen leaves results in only a 0.1% decrease in WER.

For both the development and evaluation sets, rescoring with a Kneser-Ney smoothed general trigram model (as opposed to our Good-Turing smoothed general model) results in a lower WER than the topic models [12]. The Kneser-Ney model results in a WER of 39.4% on the development set and 34.9% on the evaluation set. Therefore, while topic adaptation does result in slightly better WERs than no adaptation, future work in topic adaptation must include better smoothing techniques for models built from small amounts of training data.

4 Conclusion

Large-scale, finely tuned topic adaptation is possible and does result in a decrease in perplexity and a slight decrease in WER in the Broadcast News domain. Choosing the 20 most topic-similar clusters for an individual story from among 5883 candidates and interpolating models built from these clusters results in a 15% decrease in perplexity over a general Broadcast News model, even when the word error rate of the story hypothesis used for topic detection is quite high. Having many candidate clusters permits fine topic distinction and the possibility of mixing of topics in a way that might not have been previously seen in the training data. Furthermore, the semantic landscape of Broadcast News has been mapped out in two different topic trees. Future work may find these structures helpful in more complex topic detection and adaptation systems.

5 Acknowledgements

We would like to thank Richard Schwartz, Yiming Yang, Stanley Chen and Bin Zhou for their contributions to this work.

References

[1] B. Carlson. Unsupervised topic clustering of switchboard speech messages. In *Proceedings of ICASSP-96*, pages 315–318, 1996.

[2] R. Iyer and M. Ostendorf. Modeling long distance dependence in language: Topic mixtures vs. dynamic cache models. In *Proceedings of ICSLP*, pages 236–239, 1996.

[3] P. Clarkson and A. Robinson. Language model adaptation using mixtures and an exponentially decaying cache. In *Proceedings of ICASSP-97*, pages 799–802, 1997.

[4] G. Salton. Developments in automatic text retrieval. *Science*, 253:974–980, 1991.

[5] T. Imai, R. Schwartz, F. Kubala, and L. Nguyen. Improved topic discrimination of broadcast news using a model of multiple simultaneous topics. In *Proceedings of ICASSP-97*, pages 727–730, 1997.

[6] T. Joachims. A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. Technical Report CMU-CS-96-118, Carnegie Mellon University, March 1996.

[7] Slava M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-35(3):400–401, March 1987.

[8] Ronald Rosenfeld. The CMU statistical language modeling toolkit and its use in the 1994 ARPA CSR evaluation. In *Proceedings of the Spoken Language Systems Technology Workshop*, pages 47–50, Austin, Texas, January 1995.

[9] http://www.thomson.com/psmedia/bnews.html.

[10] I. Witten, A. Moffat, and T. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Van Nostrand Reinhold, 1994.

[11] P. Placeway et al. The 1996 Hub-4 Sphinx-3 system. In *Proceedings of the 1997 ARPA Speech Recognition Workshop*, 1997.

[12] K. Seymore, S. Chen, M. Eskenazi, and R. Rosenfeld. Language and pronunciation modeling in the CMU 1996 Hub 4 Evaluation. In *Proceedings of the 1997 ARPA Speech Recognition Workshop*, 1997.