USING STORY TOPICS FOR LANGUAGE MODEL ADAPTATION

Kristie Seymore and Ronald Rosenfeld
School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
kseymore@cs.cmu.edu, roni@cs.cmu.edu

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 WER 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. Specifically, we examine the reduction in perplexity and word error rate made possible by detecting a story's topic and then using a series of interpolated language models trained on topic-specific data to reevaluate speech recognition hypotheses. The most similar topics to a new piece of text are chosen from 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. For details of our earlier work in topic adaptation, see [1].

2. TOPIC ADAPTATION

The topic adaptation scheme we are using consists of the following three 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 the text that is being decoded.
3) Language models are built from the clusters of data found to be the most similar to the test data. The 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.

2.1. Clustering

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

Topic trees can be built by treating the topic clusters as leaves and iteratively merging the topics together to form a tree. 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, which was used for adaptation. However, one advantage of retaining a high number 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. As similar clusters are merged together, they lose their topic focus, but they acquire the advantage of having additional data to build more statistically sound language models. A topic tree is one way to combine the data advantages of larger clusters and the topic focus of many of smaller clusters. 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 use of topic trees has also been explored in the Switchboard domain by Carlson [4].

Automatic topic clustering does not always result in optimal clustering decisions. We are investigating semi-automatic methods, where the system asks for cues whenever its confidence in its clustering decision is weak. We have developed a web interface that allows the user to make clustering decisions when building a topic tree, drawing from all the text, keyword, and tree information available.

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 work, no corrective action was taken to account for the similarity measure bias due to data overlap.

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 naïve Bayes classifier. The TFIDF measure [5] assigns a weight to each unique word in a document representing how topic-specific that word is to its document or cluster. The similarity between two documents can be computed by representing each document as a vector of weights, and then computing the cosine of the angle between the two vectors. The resulting similarity measure is a value between zero and one, with zero indicating no topic correlation, and one meaning an identical match. A Naïve Bayes classifier calculates the probability of a topic given the words in a new document. In comparing these two classifiers [6], we found that the naïve Bayes classifier consistently outperforms the TFIDF classifier in both precision and recall on a Broadcast News test set where the manually assigned keywords indicate the correct classifications. See [6] for details.

2.3 Model Interpolation

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 leaf clusters. Individual language models are built from the chosen clusters (or from nodes farther up in the tree when a topic tree is being used), and the models are interpolated together at the word level. The hypothesis is then reevaluated according to the language scores of the interpolated language models. Even when the word error rate of the decoder hypothesis is significant, topic detection will still perform reasonably well [1]. As long as the word errors in the hypothesis are not significantly topic-correlated, the correct content words in the hypothesis will provide enough weight for the selection of appropriate clusters.

3. EXPERIMENTS

The training data used in these experiments is the Broadcast News corpus obtained from Primary Source Media. The data covers the period from 1992 - 1995 and consists of 130 million words. Story boundaries are marked, and each story is accompanied by a set of keywords 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.

The most frequent 63k words from the four years of Broadcast News text defined the vocabulary for calculating cluster similarity. 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 ranges from 6 to 2131.

3.1. Perplexity Reduction

In order to determine the best way to interpolate topic specific language models, we varied the number of topic specific models and measured development set perplexity. First, topic detection was run using the TFIDF and naïve Bayes classifiers on errorful first-pass Sphinx-3 [7] 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 the Linguistic Data Consortium's (LDC) release of the Broadcast News corpus. Good-Turing discounted trigram backoff language models [8] 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. Interpolation weights and perplexity values were obtained with the EM algorithm and 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 1. Using twenty topic models chosen by the naïve Bayes classifier yields the greatest reduction in perplexity from 222 with the general model to 188, a 15% reduction.

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

Table 1. Development set perplexity, leaves only

Next, we built two topic trees. The first (automatic) tree merged 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.

Of the 5883 leaf clusters, 230 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 six cases, the perplexity decreased or stayed the same when a model built from a node with more data than the leaf cluster was used. This limited example indicates that using more data by traveling up the tree from the leaf nodes may improve adaptation.

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 (automatic and orphan), whenever a leaf cluster was chosen for interpolation, the topic model was built from the lowest node in the path that had at least as many word tokens as the pre-determined threshold. 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. In the case of the orphan tree, sometimes a leaf cluster would lead straight to the root. Therefore, two orphan tree scenarios were evaluated: in the first, the leaf clusters that had fewer tokens than the threshold but were connected directly to the root were left out completely, and in the second (designated by `+leaves'), the leaf clusters were left in even if they contained fewer tokens than the threshold if a larger node with more tokens than the threshold was not available. Perplexity results for these cases are shown in Tables 2 and 3. In all cases there is a perplexity reduction over the general trigram model, and the orphan '+leaves' trees do as well as using leaves only when the leaves are chosen by the naïve Bayes classifier.

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

Table 2. 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 3. Development set perplexity, orphan tree

3.2. 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, 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 assigned a weight of 0.55 to the general 51k language model and uniform interpolation weights to the remaining topic models. In all cases, twenty leaf clusters were chosen per story. Rescoring consisted of using the original acoustic score, the new language model score, and a word insertion penalty. Filled pauses were predicted from manually set unigram probabilities [1]. 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, a WER of 40.1% was obtained. 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-3 decoder. Rescoring was tried using TFIDF-chosen leaves, Bayes-chosen leaves, and the 200k+leaves orphan tree with Bayes-chosen leaves. The two interpolation weighting schemes, minimized perplexity and uniform weights, were tested for each condition. Results are shown in Tables 4 and 5.

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, 200k orphan tree, min PP	39.6%
Bayes, 200k orphan tree, uniform	39.6%

Table 4. Development set word error rate using different language scores

Condition	WER
2^{nd} 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, 200k orphan tree, min PP	35.3%
Bayes, 200k orphan tree, uniform	35.5%

Table 5. Evaluation set word error rate

All of the topic adaptation methods lead to an improved WER on the development set, with the Bayes-chosen leaves providing the greatest WER reduction of 0.7% over 1^{st} pass development hypothesis. Improvement on the evaluation set is less significant, with most methods providing a very slight decrease in WER. The choice of model interpolation weights does not seem to significantly affect WER results, with the minimized perplexity weights performing slightly better than the uniform weights. On both the development and evaluation sets, using a Kneser-Ney smoothed general trigram model to rescore results in a lower WER than the topic models [1]. A Kneser-Ney model results in a WER of 39.4% on the development set and 34.9% on the evaluation set. 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 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. For a more detailed presentation of this work, see [6].

5. ACKNOWLEDGEMENTS

We would like to thank Richard Schwartz, Yiming Yang, Stanley Chen and Bin Zhou for their contributions and help with this work. This research was sponsored by the Department of the Navy, Naval Research Laboratory under Grant No. N00014-93-1-2005 and by the National Security Agency under Grant numbers MDA904-96-1-0113 and MDA904-97-1-0006. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The first author is additionally supported under a National Science Foundation Graduate Research Fellowship.

6. REFERENCES

[1] K. Seymore, S. Chen, M. Eskenazi and R. Rosenfeld, "Language and Pronunciation Modeling in the CMU 1996 Hub 4 Evaluation," *Proc. of the 1997 ARPA Speech Recognition Workshop*, 1997.

[2] R. Iyer, M. Ostendorf, "Modeling Long Distance Dependence in Language: Topic Mixtures vs. Dynamic Cache Models," *Proc. ICSLP*, vol. 1, 1996, pp. 236-239.

[3] P. Clarkson, A. Robinson, "Language Model Adaptation Using Mixtures and an Exponentially Decaying Cache," *Proc. ICASSP*, vol. 2, 1997, pp. 799-802.

[4] B. Carlson, "Unsupervised Topic Clustering of Switchboard Speech Messages", *Proc. ICASSP*, 1996, pp. 315-318.

[5] G. Salton, "Developments in Automatic Text Retrieval," *Science*, Vol. 253, 1991, pp. 974-980.

[6] K. Seymore, R. Rosenfeld, "Large-scale Topic Detection and Language Model Adaptation", *Carnegie Mellon University Technical Report*, June 1997.

[7] P. Placeway et al., "The 1996 Hub-4 Sphinx-3 System," *Proc. of the 1997 ARPA Speech Recognition Workshop*, 1997.

[8] S. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. ASSP-35, no. 3, March 1987, pp. 400-401.