TOPIC ADAPTATION FOR LANGUAGE MODELING USING UNNORMALIZED EXPONENTIAL MODELS

Stanley F. Chen, Kristie Seymore, Ronald Rosenfeld

School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213
{sfc,kseymore,roni}@cs.cmu.edu

ABSTRACT

In this paper, we present novel techniques for performing topic adaptation on an n-gram language model. Given training text labeled with topic information, we automatically identify the most relevant topics for new text. We adapt our language model toward these topics using an exponential model, by adjusting probabilities in our model to agree with those found in the topical subset of the training data. For efficiency, we do not normalize the model; that is, we do not require that the "probabilities" in the language model sum to 1. With these techniques, we were able to achieve a modest reduction in speech recognition word-error rate in the Broadcast News domain.

1. INTRODUCTION

A language model is a probability distribution $p(w|h)$ estimating how frequently a word w occurs given that the *history* (or previous words in the sentence) is h. Language models have many applications, most notably in speech recognition in helping to disambiguate acoustically ambiguous utterances.

The dominant technology in language modeling are n-gram models. In speech recognition, typically a single n-gram model (usually a trigram model) is built on the training data. The task of *topic adaptation* is concerned with identifying the topic of new data and adapting the language model toward that topic. For example, if a speech document is recognized as describing O.J. Simpson's trial, then the probability of the word *Kato* occurring should be boosted.

There has been much previous work in topic adaptation.[1] Numerous efforts have demonstrated large improvements in the measure of *perplexity* [2, 4, 9]; however, perplexity has been shown to correlate poorly with speech recognition performance. Several papers have reported modest speech recognition word-error rate (WER) improvements of about 0.5% absolute: Sekine and Grishman[14] add *ad hoc* topic and cache scores to their language model score in log probability space, and Iyer and Ostendorf[3] and Seymore and Rosenfeld[16] use linear interpolation to combine topic n-gram models with a general n-gram model.

In this work, we extend the research in [16] by using unnormalized exponential models to combine topic information. In [16], a first-pass transcription hypothesis is generated for each article in the test set using an unadapted trigram model. The twenty most relevant topics for each hypothesis are identified using a Bayes classifier. Then, a trigram model is built for each of these topics by just using those articles in the training data labeled with the given topic. (Each article in the training data is manually annotated with topic information.) Finally, these twenty models are linearly interpolated with a trigram model built on the entire training set to yield the language model used for speech recognition.

Recently, there has been evidence that exponential models are superior to linear interpolation in combining multiple information sources[13, 5, 4]. Exponential models have the following form

$$p(w|h) = \frac{1}{Z(h)} \exp\left(\sum_i f_i(h,w)\lambda_i\right) p_0(w|h) \qquad (1)$$

where $Z(h) = \sum_w \exp(\sum_i f_i(h,w)\lambda_i) p_0(w|h)$ is a normalization term, $p_0(w|h)$ is a *prior* probability, $f_i(h,w)$ are the *features* of the model, and λ_i are parameters associated with these features.

As an example, consider the case where we take $p_0(w|h)$ to be a trigram model. If there are no features f_i, then we will simply have that $p(w|h) = p_0(w|h)$. However, let us say that we want to model the phenomenon that the word *Kato* is more common when the topic is *O.J. Simpson*. We can do this by creating a feature

$$f_1(h,w) = \begin{cases} 1 & \text{topic}(h) = \textit{O.J. Simpson},\ w = \textit{Kato} \\ 0 & \text{otherwise} \end{cases}$$

and by setting λ_1 such that e^{λ_1} equals how many times more probable the word *Kato* becomes. This will have the effect of boosting the probability of *Kato* when the topic is *O.J. Simpson* (and consequently depressing other probabilities through the normalization term $Z(h)$), and leaving probabilities unchanged when the topic is not *O.J. Simpson*. This procedure is the basis of how we perform topic adaptation on n-gram models.

Unfortunately, the evaluation of exponential models is expensive due to the calculation of the normalization factor $Z(h)$; this calculation generally makes exponential models orders of magnitude slower than trigram models. In this research, we omit the normalization term $Z(h)$. As a result, we no longer have *probabilities* in our model but instead *scores*, and we can no longer calculate perplexities. On the other hand, our models are virtually as

This work was supported by the National Security Agency under grants 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.

[1] Here, we only discuss research where it is necessary to identify the topic of the current text automatically. This contrasts with the situation where a topic-specific adaptation text is explicitly given, as in Spoke 2 of the 1994 ARPA CSR evaluation[6].

fast as trigram models and can easily be used to calculate WER's in expensive tasks such as lattice rescoring. To prevent scores from rising above 1, we use the following formulation

$$p(w|h) = \frac{p_{\text{aux}}(w|h)}{1 + p_{\text{aux}}(w|h)}$$

where

$$p_{\text{aux}}(w|h) = \exp\left(\sum_i f_i(h,w)\lambda_i\right) \frac{p_0(w|h)}{1 - p_0(w|h)}$$

The use of the term $\frac{p_0(w|h)}{1-p_0(w|h)}$ instead of $p_0(w|h)$ maintains the property that $p(w|h) = p_0(w|h)$ when there are no features.

We consider three types of exponential features for performing topic adaptation.

- We consider features that depress the probabilities of topical words that are off-topic, *e.g.*, the word *Kato* if the topic is *Libya*. (We use the term *topical* to describe a word whose frequency depends strongly on topic, *e.g.*, the word *Kato* as opposed to the word *that*.)
- We consider features that boost the probabilities of topical words and n-grams when they are on-topic, *e.g.*, the word *Kato* or bigram *Kato Kaelin* if the topic is *O.J. Simpson*.
- We consider features that boost the probabilities of words and n-grams that occur frequently in the current article being evaluated. These features are similar in effect to a language model *cache*[7].

In the next sections, we discuss each of these feature types in turn.

Our training data consists of 121,000 articles of Broadcast News data containing a total of 130M words, with each article manually labeled with a set of topics.[2] Each article is labeled on average with ~3.6 topics out of a set of about 10,000.

2. DEPRESSING OFF-TOPIC WORD PROBABILITIES

The frequency of a topical word in off-topic articles will often be much lower than its frequency calculated over the entire training set. For example, in 130M words of Broadcast News text, the word *Kato* occurs 3111 times, yielding a unigram frequency of about 2.4×10^{-5}. However, 2990 of these occurrences happen within articles labeled with the topic *O.J. Simpson*, these articles comprising a total of 16M words. Thus, the word *Kato* has a frequency of only $\frac{3111-2990}{(130-16)\times 10^6} \approx 1.1 \times 10^{-6}$ when the topic is not *O.J. Simpson*, which is more than ten times less than its general frequency.

Modeling this phenomenon in an exponential model is fairly straightforward: referring to equation (1), we want to find a factor λ_w for each word w such that e^{λ_w} expresses how much less frequently that word occurs in off-topic text than in general text, *i.e.*,

$$e^{\lambda_w} = \frac{p_{\text{off-topic}}(w)}{p_0(w)} \tag{2}$$

The corresponding features f_w are of the form

$$f_w(h, w') = \begin{cases} 1 & w \text{ is off-topic w.r.t. } h,\ w' = w \\ 0 & \text{otherwise} \end{cases}$$

[2] The text and topic labels were acquired from Primary Source Media.

CARRERE	178.55
RIBERA	101.49
MADYUN	71.33
HAILES	60.52
BRANDIS	49.72
GEMCO	43.89
⋮	

Table 1: Estimates of how much less frequent words w are when off-topic (*i.e.*, $\frac{1}{e^{\lambda_w}}$)

To calculate $p_{\text{off-topic}}(w)$ for a word w, we need to determine which topics are off- and on-topic with respect to w. One reasonable heuristic for guessing that a topic is on-topic is if the frequency of w in articles labeled with that topic is much higher than its frequency over the entire training set. However, this heuristic is not ideal as indirect dependencies may exist. For example, if many articles with the topic *O.J. Simpson* are also labeled with the topic *DNA testing* (recall that articles usually have multiple topics), then the topic *DNA testing* may be considered on-topic for the word *Kato* according to this heuristic.

A method for modeling these partial dependencies is to use *maximum entropy* training for exponential models[1]. Consider a *topic unigram* model, or model with features of the forms

$$\begin{aligned} f_{T,w}(h, w') &= \begin{cases} 1 & T \in \text{topic}(h),\ w' = w \\ 0 & \text{otherwise} \end{cases} \\ f_w(h, w') &= \begin{cases} 1 & w' = w \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{3}$$

for each topic T and word w. (For p_0 in equation (1), we use a uniform distribution.) After maximum entropy training, the magnitude of each parameter $\lambda_{T,w}$ will be, roughly speaking, an indication of how strongly correlated the word w is with topic T, taking into account indirect dependencies. Furthermore, $p(w|h)$ for a history h where *topic*$(h) = \epsilon$ is an estimate of the frequency $p_{\text{off-topic}}(w)$ we need in equation (2).

The complete procedure we used to calculate our off-topic depression factors is as follows: we began with a 51k vocabulary of the most common words in the Broadcast News data. To reduce the number of features in the topic unigram model to a manageable size, we only included the feature $f_{T,w}$ if the word w occurred much more frequently in articles labeled with topic T than in general according to a χ^2 test. This process yielded about 200,000 features. Unlike the other exponential models used in this work, the topic unigram model was normalized. We used optimizations as described by Lafferty and Suhm[8] in the maximum entropy training; each iteration took less than 10 minutes on a Pentium II processor. The training yielded positive depression factors for 30,000 words. An excerpt of these factors is displayed in Table 1.

In evaluation, we used the procedure described in Section 1 to find twenty relevant topics for each article. We took a word w to be off-topic if the frequency of w in the training data in each of the twenty topics was not significantly higher than its off-topic unigram probability according to a χ^2 test.

3. BOOSTING ON-TOPIC N-GRAM PROBABILITIES

In boosting the probabilities of words and n-grams that are topical and on-topic, first consider the case where we would like to adapt a language model toward a *single* topic T. A reasonable procedure would be to set each adapted probability $p_{\text{adapt}}(w|h)$ to the baseline n-gram probability $p_0(w|h)$ unless the topic probability $p_T(w|h)$ is significantly different (*e.g.*, according to a χ^2 test), in which case the adapted probability should be set to the topic probability. We can take the topic model $p_T(w|h)$ to be an n-gram model built on the training data labeled with topic T.

To perform this adaptation for exponential models, we can first loop through all unigrams w. Whenever $p_T(w)$ is significantly different from $p_0(w)$ we add a feature $f_w(h, w')$ as in equation (3) with λ_w set such that $e^\lambda = \frac{p_T(w)}{p_0(w)}$. Then, we loop through all bigrams $w_{i-1}w_i$, comparing $p_T(w_i|w_{i-1})$ against $p_0(w_i|w_{i-1})$ combined with all unigram features created. (In exponential models, an n-gram feature affects all n'-gram probabilities for $n' \geq n$.) We can repeat this process for all levels of the n-gram model.[3]

However, articles are generally a combination of multiple topics, and it is not clear how to reconcile probabilities in this more complex situation, especially in light of the indirect dependencies mentioned in Section 2. A theoretically motivated method would be to build a maximum entropy topic n-gram model (analogous to the topic unigram model described earlier) and to train this model on the entire training set; however, this would require a stupendous amount of computation.

We instead choose a simple heuristic that can be considered in spirit to be a very poor approximation to maximum entropy training. In particular, for each level of our n-gram model we apply the procedure described previously for adapting to a single topic to each of the topics in turn, except that we only consider probability *increases*. That is, for each probability $p_{\text{adapt}}(w|h)$ we take the maximal $p_T(w|h)$ over all of the relevant topics T, as long as this probability is significantly higher than the baseline n-gram probability according to a χ^2 test. Intuitively, we are assuming that the probability of a word or n-gram in the adapted model should be large if it is large in *any* of the relevant topics.

3.1. Filtering Adaptation Topics

We have found that usually not all of the twenty topics for an article returned by our Bayes classifier are relevant. To select the most relevant topics of the twenty, we build a model for each topic adapting the general model to just that topic. We calculate the likelihood of the first-pass hypothesis transcription using these models, and use a topic only if its corresponding likelihood is substantially lower (0.3 bits/word) than the likelihood assigned by the general model.[4] In Table 2, we display the results of this process for an article concerning racial issues between blacks and whites.

3.2. Boosting Article-Specific n-Gram Probabilities

Cache models attempt to characterize the phenomenon that words and n-grams tend to repeat themselves within articles, by increasing the probabilities of n-grams that have occurred previously in an article[7]. We can place this type of modeling within our adaptation framework by viewing the first-pass hypothesis transcription of an article to be another topic adaptation text. We can adapt our language model to this text in the same way that we adapt it to each relevant topic. Words or n-grams that occur surprisingly frequently in the hypothesis will have their probabilities boosted in the adapted language model.

In conventional caching, hypotheses are processed beginning-to-end and all previous words in a hypothesis are assumed to be correct and placed in the cache. In our scheme, the whole article is processed before features are created, and features are only created if they pass a significance test. Thus, it seems likely that our scheme is less susceptible to speech recognition errors.

kept	filtered out	
Racism	*Murder*	*Political_activity*
Blacks	*Presidents*	*Criminal_justice*
Race_discrimination	*Clinton,_Bill*	*Administration*
Minorities	*United_States*	*Race_relations*
Prejudice	*Social_conditions*	
Employment	*Economic_conditions*	
Discrim.,_employment	*Crime_and_criminals*	
Affirmative_action	*Politics_and_government*	

Table 2: Results of topic filtering by likelihood for an article concerning racial issues between blacks and whites

4. EXPERIMENTS

In our experiments, we used speech recognition lattices generated by the Sphinx-III system[10] on 20 articles of Broadcast News data (16,700 words). For each article, we first generated a hypothesis using a trigram model generated by the CMU language modeling toolkit[11] from our 130M words of training text. The word-error rate of these hypotheses were 30.8%. We found twenty relevant topics for each article using a Bayes classifier on these first-pass hypotheses. In each experiment, word-error rates were calculated through lattice rescoring with the adapted model. The baseline model for adaptation is the trigram model described above.

4.1. Depressing Off-Topic Word Probabilities

We investigated whether the depression of off-topic word probabilities alone would improve word-error rate. Using the 30,000 depression features described in Section 2, we found that the WER improved by 0.1% absolute to 30.7%. To get a detailed view of the variation between the hypothesis generated by the baseline trigram model and the hypothesis generated by the adapted model, we aligned these two hypotheses to find their word differences. We then aligned these differences against the reference transcript, to determine how many errors were fixed and created with the adapted model. Over the 16,700 words in the test set, there were 43 word differences between the baseline and adapted hypotheses. Of these 43 differences, 17 were errors fixed in the adapted hypothesis, 5 were errors created, and 21 were errors in both hypotheses.

As an upper bound on the WER reduction of these techniques, Rosenfeld *et al.*[12, 15] estimate that if no out-of-vocabulary errors are introduced, then removing 10,000 words from a large vocabulary improves WER by about 0.2% absolute, so depressing 30,000

[3]This procedure is a crude but quick approximation to maximum entropy training with this feature set. It would be more sound (but vastly more expensive) to set the parameters λ using a true maximum entropy training algorithm.

[4]Because calculating an exact likelihood would be expensive due to normalization costs, we use approximations to calculate the likelihood.

art.	no. words	base WER	topic adapt	art. adapt	both adapt	unig. adapt
A	1724	37.1%	36.0%	36.3%	35.3%	35.8%
B	2761	34.0%	34.0%	34.1%	34.1%	34.2%
C	3499	30.3%	30.3%	30.2%	30.1%	30.4%
D	2529	37.7%	38.2%	37.5%	38.2%	38.1%
E	3928	26.5%	26.1%	26.3%	25.7%	26.1%
F	2259	22.3%	22.0%	21.6%	21.4%	21.3%
tot.	16700	30.8%	30.6%	30.5%	30.3%	30.5%

Table 3: Speech recognition performance for models with on-topic and article-specific n-gram features

words completely and perfectly would lead to a WER improvement of about 0.6%.

4.2. Boosting On-Topic and Article-Specific n-Gram Probabilities

In experiments with on-topic and article-specific features, we did not use depression features as they seemed to have little effect. We performed adaptation with unigram and bigram features. We display the article-by-article error rates of on-topic and article-specific adaptation in Table 3. We achieved our best WER improvement of 0.5% absolute using both adaptations together. Improvements varied widely between articles, with our best article WER improvement being 1.8% absolute in article A. In the final column of the table, we display the results of adding only unigram adaptation features; bigram features seem to effect a small improvement.

Comparing the baseline and best adaptation hypotheses using the methodology described in Section 4.1, we found that the two hypotheses differed by 854 words. Of these 854 words, 261 were errors fixed by adaptation, 162 were errors created by adaptation, and 431 were errors in both hypotheses.

5. DISCUSSION

To summarize, we introduced several novel topic adaptation techniques for unnormalized exponential models. The use of unnormalized exponential models has the advantage of efficient computation while hopefully retaining some of the properties of conventional exponential models. We were able to run lattice rescoring experiments at about 3 times real-time on a Pentium II processor. Because we use unnormalized models, it is not meaningful to calculate perplexity; however, perplexity has been shown to correlate poorly with speech recognition performance.

This work is the first to explicitly model the depression of off-topic word probabilities. We describe how to use maximum entropy training to determine these depression factors. We present a novel implementation for robust caching, which fits in a unified manner within our topic adaptation framework. We describe an effective method for filtering out irrelevant topics by using the likelihood of the first-pass transcription. Throughout our work, we use statistical testing to select only those adaptation features which are significant.

We achieved a minimal reduction in WER by depressing off-topic word probabilities, but achieved a modest reduction through boosting on-topic and article-specific n-gram probabilities. Our WER reduction is comparable to the best existing results for this task.

6. REFERENCES

[1] A. Berger, S. Della Pietra, and V. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[2] P. Clarkson and A. Robinson. Language model adaptation using mixtures and an exponentially decaying cache. In *Proc. ICASSP-97*, 1997.

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

[4] R. Kneser and J. Peters. Semantic clustering for adaptive language modeling. In *Proc. ICASSP-97*, volume 2, pages 779–782, 1997.

[5] R. Kneser, J. Peters, and D. Klakow. Language model adaptation using dynamic marginals. In *Proc. Eurospeech '97*, 1997.

[6] F. Kubala. Design of the 1994 CSR benchmark tests. In *Proc. Spoken Language Sys. Technology Workshop*, pages 41–46, January 1995.

[7] R. Kuhn and R. D. Mori. A cache-based natural language model for speech reproduction. *IEEE Trans. PAMI*, 12(6):570–583, 1990.

[8] J. Lafferty and B. Suhm. Cluster expansions and iterative scaling for maximum entropy language models. In K. Hanson and R. Silver, editors, *Maximum Entropy and Bayesian Methods*. Kluwer Academic Publishers, 1995.

[9] S. Martin, J. Liermann, and H. Ney. Adaptive topic-dependent language modelling using word-based varigrams. In *Proc. Eurospeech '97*, 1997.

[10] P. Placeway, S. Chen, M. Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K. Seymore, M. Siegler, R. Stern, and E. Thayer. The 1996 Hub-4 Sphinx-3 system. In *Proc. DARPA Speech Recog. Workshop*, February 1997.

[11] R. Rosenfeld. The CMU statistical language modeling toolkit and its use in the 1994 ARPA CSR evaluation. In *Proc. Spoken Language Sys. Technology Workshop*, pages 47–50, Austin, Texas, January 1995.

[12] R. Rosenfeld. Optimizing lexical and n-gram coverage via judicious use of linguistic data. In *Proc. Eurospeech '95*, pages 1763–1766, 1995.

[13] R. Rosenfeld. A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech, and Language*, 10, 1996.

[14] S. Sekine and R. Grisham. NYU language modeling experiments for the 1995 CSR evaluation. In *Proc. ARPA Spoken Language Sys. Technology Workshop*, 1995.

[15] K. Seymore, S. Chen, M. Eskenazi, and R. Rosenfeld. Language and pronunciation modeling in the CMU 1996 Hub 4 evaluation. In *Proc. DARPA Speech Recog. Workshop*, Washington, D.C., February 1997.

[16] K. Seymore and R. Rosenfeld. Using story topics for language model adaptation. In *Proc. Eurospeech '97*, 1997.