Measuring convergence in language model estimation using relative entropy

Abhinav Sethy, Shrikanth Narayanan

Speech Analysis and Interpretation Lab
Integrated Media Systems Center
Viterbi School of Engineering
Department of Electrical Engineering-Systems
University of Southern California

Bhuvana Ramabhadran

Human Language Technologies
IBM T. J. Watson Research Center
Yorktown Heights, NY

Abstract

Language models are generally estimated using smoothed counting techniques. These counting schemes can be viewed as non linear functions operating on a Bernoulli process which converge asymptotically to the true density. The rate at which these counting schemes converge to the true density is constrained by the training data set available and the nature of the language model (LM) being estimated. In this paper we look at language model estimates as random variables and present an efficient relative entropy (R.E) based approach to study their convergence with increasing training data size. We present experimental results for language modeling in a generic LVCSR system and a medical domain dialogue task. We also present an efficient recursive R.E computation method which can be used as a LM distance measure for a number of tasks including LM clustering.

1. Introduction

Language models are generally conceived as probability models which define the probability of seeing a word sequence W in a sentence. The most common language models are specified in terms of a hierarchical sequence of n-grams which specify the probability of observing a word w given the sequence of $n-1$ words specified by the history h which have been observed before it. These models are simple to implement in terms of model parameter estimation and calculation of the probability of a word sequence W.

Given a set of utterances for training, a n-gram language model can be generated using a simple counting process which counts the number of times (k) a word w is seen in training data after the history h. If the history h occurs N times in the training data, the maximum likelihood estimate of probability $P(w/h)$=k/N. However this estimate is itself a random variable which can be shown to converge to the true $P(w/h)$ as N increases to infinity. Under different constraints we can approximate this estimate by a Gaussian or a Laplacian with mean equal to the true probability.

Although asymptotically convergent, availability of training data limits the use of the maximum likelihood based counting scheme. A large number of history and word pairs are either not seen at all in the training corpus or their counts (N, k) are very low which means that the LM estimate will have a large variance. A number of smoothing techniques[2] exist to distribute the probability mass among the n-grams in a more uniform fashion, so that the estimated model is more accurate.

The convergence rate of a language model to the true density is an important consideration in determining the type of language model one should use for given training data constraints. Some of the factors directly effected by the confidence we can place on the LM estimates include the LM n-gram order, selection of LM classes etc. In addition, it effects the weighting a language model receives in various classifier combination tasks such as speech recognition, topic detection etc.

One of the popular measures to evaluate the goodness of language models is perplexity. Perplexity is defined as $2^{H(T)}$ where $H(T)$ is the cross entropy of a given test set T

$$H(T) = -\frac{1}{k}\sum_{i=1}^{k} \log_2 p(w_i/h_i)$$

Perplexity is essentially a measure of the match between the language model and a held out test set T. Previous work[2][8] has compared performance of smoothing schemes across increasing data size in terms of perplexity measurements.

Another distance metric that can be used to compare two discrete probability models such as language models is relative entropy (R.E) or Kullback Leibler distance which is defined as

$$D(p,q) = \sum_{x\in X} p(x) \ln \frac{p(x)}{q(x)}$$

If the estimated language model matches the true density closely, R.E between them is low and is zero, if they match exactly. R.E can be seen as the difference between the entropy of the true density and the cross entropy of the estimated LM on infinite test data. In this paper we will present a technique (Sec. 2) for convergence analysis of language models based on R.E and show that it provides supplemental information to perplexity. To enable this we develop an efficient implementation (Sec. 3) of R.E for n-gram language models which can be extended to a number of other LM tasks (Sec. 5). Results and experimental details are provided in section 4. We conclude with an analysis of our results and directions for future work.

2. LM convergence analysis using R.E

In order to specify how close an estimated LM is from the true density we need to define a distance measure. Perplexity[3][7] can be a useful tool for this purpose. We can compute the perplexity of different language model estimates and select the one which gives lowest perplexity on the test set as the closest match to the true density. However it is difficult to interpret the results since the lower bound of perplexity is not known beforehand. A decrease in perplexity although indicative of progress does not measure how far we are from the true density. Another issue is

that the test set is a small sample drawn from the true density and does not cover the entire range of n-gram probabilities in the true language model. Also since it is essentially a random sample from the true density, it does not represent the true density accurately. If the true density is known, R.E of the LM estimate can address the issues with perplexity based analysis. R.E can provide local information in terms of convergence of different probability estimates as well as a global match between the true density and the approximating LM.

To study convergence of LM estimates we substitute the hidden true language model with a representative language model. Next we generate training data from this language model by a process that can be seen as a random walk through a graphical model which encapsulates the LM. The generated data represents a sample drawn from the specified language model. Language models can then be estimated from this 'artificial' data and R.E comparisons with the initial representative language model can be carried out. This scheme can be seen as a distribution resampling approach. The size of the 'artificial' training data can be increased to see how the R.E converges to zero. An advantage of this scheme is that it is possible to generate a training corpus of arbitrary size without effecting the underlying true data density. In conventional schemes the data available for training is limited and to study convergence we can only decrease the training set size. R.E comparisons across different classes of the LM such as proper nouns, functional words, domain specific terms can be carried out separately to estimate the confidence that can be placed in the LM estimates for these groups. In the next section we will present a method for computing R.E between n-gram language models efficiently.

3. Computation of relative entropy

Computing relative entropy between discrete distributions requires density comparisons across all the possible symbols in the alphabet. This implies that a direct R.E implementation for a n-gram language models would require V^n computations, where V is the vocabulary size. This would make R.E comparisons for even medium sized trigram LM's with 15-20K words computationally prohibitive.

However real world n-gram language models are tree structured and a lot of the n-gram densities backoff to probabilities of corresponding $n-1$ grams. Based on this tree like structure, we provide a scheme which makes it possible to compute R.E between two LM's in $O(L)$ computations where L is the number of language model terms actually present in the two LM's. We will express the language models as $p(x/h)$ and $q(x/h)$ where h is the history on which the probability of seeing the word x is conditioned; p being the reference LM and q being the LM being evaluated with respect to p. In case of convergence analysis q is the language model estimated from artificial data.

The other symbols we are going to use are:

x: The current word
h: The history $w_1..w_{n-1}$
h': The back off history $w_2..w_{n-1}$
b_h: The back-off weight for p distribution for history h
b'_h: The back-off weight for the q distribution
W: The vocabulary of the language model

R.E at level n

$$D_n = \sum_{h \in H} p_h \sum_{x \in W} p(x/h) \ln \frac{p(x/h)}{q(x/h)} \tag{1}$$

We can divide the set of histories (H) at level n into H_s for all h which exist as $n-1$ gram and have a back-off weight$\neq 1$ in the p or the q distribution. The complement set ($H_{s'}$) will contain histories with a back-off 1. $H_{s'}$ corresponds to histories not seen in either language model. Let

$$D_{xh} = \sum_{x \in W} p(x/h) \ln \frac{p(x/h)}{q(x/h)} \tag{2}$$

Then

$$D_n = \sum_{h \in H_s} p_h D_{xh} + \sum_{h \in H_{s'}} p_h D_{xh}$$

$$= \sum_{h \in H} p_h D_{xh'} + \sum_{h \in H_s} p_h D_{xh} - \sum_{h \in H_s} p_h D_{xh'}$$

Marginalizing w_1

$$D_n = D_{n-1} + \sum_{h \in H_s} p_h \left(D_{xh} - D_{xh'} \right) \tag{3}$$

D_{xh} can be split into four terms depending on whether x/h is defined in the p or the q distribution

D_{xh}=$T_1 + T_2 + T_3 + T_4$
T_1: $p(x/h)$ exists $q(x/h)$ backs-off (Let $x \in X_1$)
T_2: $p(x/h)$ backs-off $q(x/h)$ exists (Let $x \in X_2$)
T_3: $p(x/h)$ exists $q(x/h)$ backs-off (Let $x \in X_3$)
T_4: $p(x/h)$ backs-off $q(x/h)$ backs-off (Let $x \in X_4$)

$$T_1 = \sum_{x \in X_1} p(x/h) \ln \frac{p(x/h)}{q(x/h)}$$

$$T_2 = b_h \ln b_h \sum_{x \in X_2} p(x/h') + b_h \sum_{x \in X_2} p(x/h') \ln \frac{p(x/h')}{q(x/h)}$$

$$T_3 = \sum_{x \in X_3} p(x/h) \ln \frac{p(x/h)}{q(x/h')} - \ln b'_h \sum_{x \in X_3} p(x/h)$$

$$T_4 = \sum_{x \in X_4} b_h p(x/h') \ln \frac{b_h p(x/h')}{b'_h q(x/h')}$$

$$= b_h \ln \frac{b_h}{b'_h} \sum_{x \in X_4} p(x/h') + b_h \sum_{x \in X_4} p(x/h') \ln \frac{p(x/h')}{q(x/h')}$$

$$+ b_h \sum_{x \in X'_4} p(x/h') \ln \frac{p(x/h')}{q(x/h')} - b_h \sum_{x \in X'_4} p(x/h') \ln \frac{p(x/h')}{q(x/h')}$$

$$= b_h \ln \frac{b_h}{b'_h} \left(1 - \sum_{x \in X'_4} p(x/h') \right) + b_h D_{xh'}$$

$$- b_h \sum_{x \in X'_4} p(x/h') \ln \frac{p(x/h')}{q(x/h')}$$

Thus we are able to express D_{xh}, in terms of the LM terms actually seen. Using D_{xh} computed in this fashion in (3) we get a recursive formulation for R.E at level n using LM densities actually seen. D_{xh} can be computed using the base expression(2) if $p(x/h)$ or $q(x/h)$ are mostly defined in the two LM's for the history h.

In the next section we describe our experimental setup for using this recursive technique in measuring LM convergence.

4. Results

4.1. Reference language models

Our experiments were carried out using the SRI toolkit[5]. We applied the resampling technique (Sec. 2) on two different language models: An interpolated trigram LM for LVCSR based on conversational speech(LM1) and a task adapted medical domain trigram language model(LM2). LM1 [9]was built from manually transcribed 180 hours of conversational speech (1.7M words), interpolated with language models built from Broadcast News and Switchboard corpora (158M and 3.4M words, respectively) . LM2 has a vocabulary of around 20K words with 700K bigrams and 8M trigrams. This LM is used in the English part of the transonics speech to speech translation system [6]. It was built on a conversational trigram LVCSR LM adapted on semi-structured patient doctor dialog data from a variety of sources.

4.2. Convergence analysis results

We generated 'artificial data' from the two LMs and built language models on the generated data using Kneser-Ney smoothing. The estimated LM and the reference LM were compared using the recursive R.E measure (Sec. 3). We also carried out LM estimation on multiple data sets of same size to measure the variance of R.E. Variance of R.E was around 1% which implies that R.E is consistent across different runs of the resampling process. The plots (See Fig. 1,2) compare well with the results of [2].

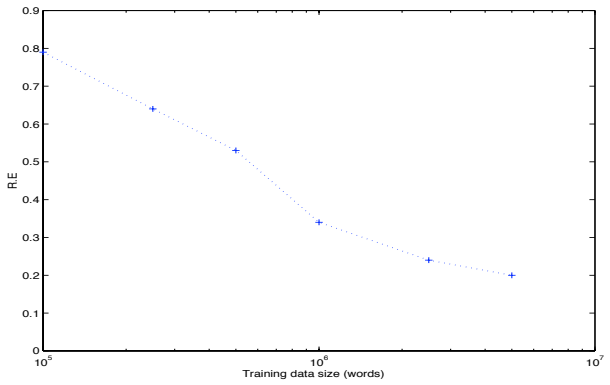

Figure 1: R.E vs training data size for LM1

For the two LMs, perplexity was also measured for the resampled language models on held out data. As expected the perplexity plots are similar to the R.E plots (See Fig. 3,4).

To observe how the language model convergence rate compared across different word categories we estimated the in-class R.E across different LM subsets grouped according to their most common part of speech tag. As can be seen in table1 the convergence of the LM to the reference varies widely across the different word categories. Convergence for nouns (specifically proper nouns) was very poor compared to the functional words such as verbs, adjectives etc. This would imply that in weighted classifier merging a low LM weight should be chosen for nouns and a higher weight for functional words. In addition class based models could be advantageous for these words in terms of estimation accuracy.

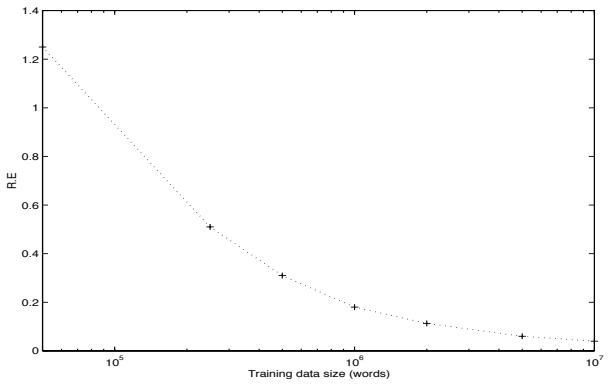

Figure 2: R.E vs training data size for LM2

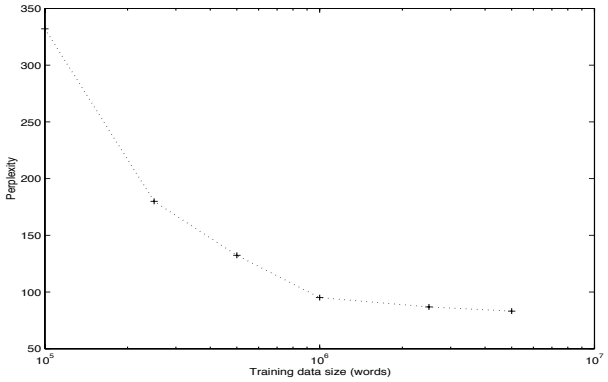

Figure 3: Perplexity vs. training data size for LM1

4.3. Sparseness of resampled LMs

As can be seen in table 2, resampled LMs were much sparser than the reference LM (LM2) even though the R.E and corresponding perplexity figures were very low. The reference LM has a perplexity of 11.39 with 8M trigrams.

LM pruning is one example application where LM reestimation and recursive R.E computation can prove useful. In the next section we discuss this in more detail.

5. Other applications of recursive R.E computation

The recursive R.E computation scheme described in section 3 can be used to compare any two nested n-gram models which

Parts of speech	Average R.E
Nouns	0.8
Proper nouns	3.4
Verbs	0.4
Adjectives	0.6
Determinants	0.3

Table 1: R.E across different parts of speech for LM2 corresponding to artificial training data size of 20M words

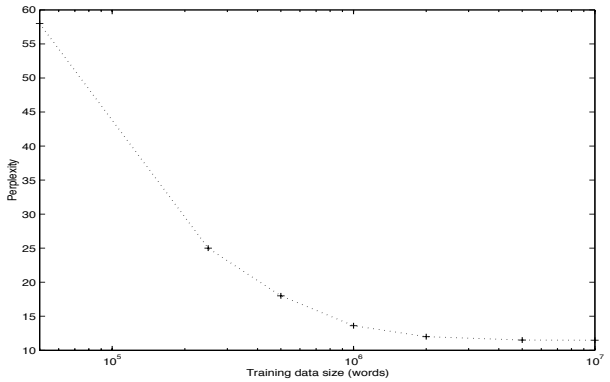

Figure 4: Perplexity vs. training data size for LM2

Generated data (K words)	R.E	Perplexity	Number of trigrams
500	0.31	18	80K
1000	0.18	13.6	118K
2000	0.11	12	188K
5000	0.06	11.5	284K
10000	0.04	11.47	464K
20000	0.02	11.44	774K

Table 2: R.E, perplexity and number of trigrams across LMs resampled from increasing generated data for LM2.

can even have different order. To address the issue of the vocabulary mismatch between two models, the words not seen in either language model can be mapped to the unknown word ($<$ unk $>$). Class based models can also be compared with minor modifications to the recursive formulation given in section 3.

A simplified recursive R.E scheme for the case when a n-gram is being approximated by a (n-1) gram was used in [4] to prune language models. Using the recursive R.E scheme, this idea can be extended to allow adjustments of the lower level probability estimates to provide a better fit for the n-gram. We also observed (Sec. 4) that resampling a LM using the generation and estimation process generates language models which are substantially sparser than the reference LM. Since the R.E measures the fit to the reference density we can use it as a guide for estimating complexity vs. accuracy tradeoff in the resampling scheme. In an information theoretic sense, this can be seen as a quantization process where the R.E measures the bits we are losing with respect to the reference LM.

Measurements of LM convergence can help in selecting the vocabulary word or histories that can be combined in one class to increase estimation robustness with minimal hit on accuracy. Alternatively the RE criteria can be used to induce a split in the training data in a decision tree like fashion to condition the language model on topic or meta-data[10].

6. Conclusion

In this paper we presented a relative entropy based approach for comparing language model estimation techniques and their convergence properties using resampling. To make R.E comparisons between language models computationally efficient we introduced a recursive R.E computation algorithm which utilizes the tree structure of n-gram language models. As described in the previous section this algorithm can be extended to compare language models with different order and vocabulary.

We plan to merge the LM generation and estimation processes to speed up the reestimation cycle. Instead of generating an artificial data corpus and then estimating the resampled LM using smoothed counting, we can combine the two processes and generate resampled LMs directly.

Our experiments indicate that for a small tradeoff in R.E, we can generate language models which are significantly sparser. Thus, resampling seems to be a promising LM pruning strategy. We are also evaluating the effectiveness of recursive R.E computation as an LM distance measure for applications like LM clustering and adaptation.

7. Acknowledgments

This research was funded in parts by the Integrated Media Systems Center, an NSF ERC, under cooperative agreement No. EEC-9529152 and by the Department of the Army under contract number DAAD 19-99-D-0046.

8. References

[1] Cover, Thomas M and Joy A. Thomas, *Elements of Information Theory*. John Wiley.

[2] Stanley Chen and Joshua T. Goodman, "An empirical study of smoothing techniques for language modeling", Proceedings of the 34th Annual meeting of the ACL, Santa Cruz, Irvine, California, 1996.

[3] Stanley Chen, Douglas Beeferman, Ronald Rosenfeld, "Evaluation Metrics For Language Models", DARPA Broadcast News Transcription and Understanding Workshop, 1998.

[4] Andreas Stolcke, "Entropy-based Pruning of Backoff Language Models", DARPA Broadcast News Transcription and Understanding Workshop, 1998.

[5] Andreas Stolcke, "SRILM - An Extensible Language Modeling Toolkit", Proc. Intl. Conf. Spoken Language Processing, Denver, Colorado, 2002.

[6] S. Narayanan, S. Ananthakrishnan et al., "Transonics: A speech to speech system for English-Persian Interactions", Proc. IEEE ASRU, St. Thomas, U.S. Virgin Islands, December, 2003.

[7] Salim Roukos, "Language Representation, in Survey of the State of the Art in Human Language Technology", ed. Giovanni Battista Varile and Antonio Zampolli, 1995. http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html

[8] Chengxiang Zhai, John Lafferty, "A study of smoothing methods for language models applied to Ad Hoc information retrieval", Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, New Orleans, Louisiana, 2001.

[9] Bhuvana Ramabhadran, Jing Juang and Michael Picheny, "Towards Automatic Transcription of Large Spoken Archives - English ASR for the MALACH project", ICASSP, Hong Kong, 2003.

[10] M. Bacchiani and B. Roark, "Meta-data Conditional Language Modeling", ICASSP, Montreal , May 2004.