Error-Responsive Modifications to Speech Recognizers: Negative N-grams

L. Chase, R. Rosenfeld, W. Ward

Carnegie Mellon University
chase@cs.cmu.edu

ABSTRACT

We describe an error analysis technique that facilitates blame assignment among the various components of a speech recognizer and provides insight into their behavior. Tools are presented that help clarify how each of the component models and their interactions contribute to the bottom line performance. We use this technique to study the performance of the backoff [4] language model. The analysis highlights the significant effect of *negative n-grams* - sequences of words not seen in the training data. This leads to two modifications to the decoder, both of which are presented with experimental results. The first modification failed so far to improve recognition performance. The second yields up to 4% reduction in word error rate.

1. A NEW ERROR ANALYSIS TOOL

The current practice of using WER as the standard metric of performance for speech recognition systems, while useful as a common currency and as a hard bottom line for performance measurement, is limited in its ability to provide insight into the behavior of complex recognition systems. The causes of error in a large vocabulary speech recognition system such as Sphinx-II [1] fall roughly into one of five categories: problems with dictionary pronunciations, inaccuracy in the acoustic models, inaccuracy in the language model, search errors, and interactions between the component acoustic and language scoring facilities. If we only look at WER performance in evaluating our systems we will be unable to understand how each of the component models and their interactions contribute to the bottom line error performance. Also, WER captures only limited information about where on the time line errors occur and how they relate temporally to the correct decoding of the reference transcript.

In the example in Figure 1 the decoder has output two words to accomodate one word in the transcript. The begin and end times of the error actually correspond to the begin and end boundaries of the missed word in the input stream. While it may be "fair" to count two errors in this situation, it is not particularly helpful; nor is it likely that a high level summary of insertions, deletions, and substitutions that contain similar patterns will help us to understand how to modify the recognizer.

```
REF: richard SARAZEN *** chief  ...
HYP: richard SIZE    AND chief  ...
```

Figure 1: Decoder behavior for the out-of-vocabulary (OOV) word "Sarazen".

A simple step toward adjusting the WER metric for this time-related problem is to use phonological alignment (see Figure 2). Although this adjustment does straighten out the incorrect labeling of an insertion and a substitution, the larger problem of understanding exactly what caused the error sequence in the first place remains. Also, words are too coarse grain a level to capture the details of how acoustic models fail and interact with the language model.

Our analysis technique, which does address the need to identify the underlying source of errors, is based on three modular phased operations called FORCE, ERRCMP, and ERRCNT. The first is dependent on the particular decoder being analyzed, and would have to be replicated independently at a site adopting the use of the tool. ERRCMP and ERRCNT, however, are independent of the decoder being used and are available for distribution.[1]

1. *Segmentation Database (FORCE):* For each utterance a database entry is created that includes utterance name, start and end frames for each component word segment, an acoustic score for each segment, a language model score for each segment, and the type (source) of the language model score from the backoff language model. Segmentation information for both the (forced alignment of the) reference sentence and the decoder hypothesis is included.[2]
2. *Error Region Database (ERRCMP):* A set of *error regions* on the time line are defined by comparing the segmentations of the reference sentences and the decoder output from the segmentationn database. Within error regions additional subregions that "explain" the same input data frames are identified. (See Figure 3 for an example of the structure imposed by this analysis on the output of the first phase.) Both machine- and human-readable databases are produced.
3. *Useful Statistics (ERRCNT):* The error region database is read by a program that can count and report any statistic relating error region identity and frame or scoring information.

2. ANALYSIS OF THE BACKOFF LANGUAGE MODEL

With the error region database in hand it is possible to ask detailed questions about the behavior of the component lexical, acoustic, and language models and about their interactions. The Katz-style [4] backoff language model implemented in Sphinx-II and many similar

[1] For a complete description of these tools and the formats of the databases they produce and use, see [2].

[2] In our current implementation the segmentation database does not include forced alignments of reference transcripts that contain OOV words w.r.t. our decoder dictionary.

Simple alignment produces:

```
REF: in the second month DR. SHENAUGH LEFT  FOR ARGENTINA ***********
HYP: in the second month DOG AND       SHEEN ALL EFFORT    ARGENTINA'S
```

Phonological alignment produces:

```
REF: in the second month DR. *** SHENAUGH LEFT  FOR    ARGENTINA
HYP: in the second month DOG AND SHEEN    ALL   EFFORT ARGENTINA'S
```

Figure 2: A problem with simple alignment that is corrected with phonological alignment.

systems will assign a finite maximum likelihood (MLE) score to any sequence of tokens from the decoder, making it possible for the decoder to produce seemingly unlikely sequences of words, especially if the acoustic match for these words is good. The idea that somehow the decoder could exploit "negative n-grams" to prevent the decoder from assigning survivable scores to errorful word sequences that did not appear in the trigram database was approached using our new error analysis tool.

Date	#Utts	#Speak	#Words	#OOV	OOV Rate
Nov93	503	10 (5M,5F)	8227	248	3.0%

Table 1: The si_dt_20 test data used in this paper.

Figure 4 reports the backoff rates in complete utterances in the si_dt_20 test set [5], described in Table 1. The backoff language model calculation of a conditional probability estimate $P(w_3|w_1, w_2)$ will fall into one of five categories [4]:

1. *T:* The trigram (w_1, w_2, w_3) is present in the language model database.
2. *BB-B:* The bigram (w_1, w_2) is present and the bigram (w_2, w_3) is present.
3. *B:* Only the bigram (w_2, w_3) is present.
4. *BB-BU:* Only the bigram (w_1, w_2) is present.
5. *BU:* Neither (w_1, w_2) nor (w_2, w_3) is present.

It is interesting to note in Figure 4 the sharp increase in the "BU" cases among OOV utterances. This sharp rise corresponds to the fact that on average 1.85 words are used to "decode" OOV words in this test set.

Figure 5 reports the relative rates of backoff types within the error regions in the test set. The number of trigrams used by the decoder in error regions drops more than 15%. This loss of trigram use spreads itself out over the four backoff cases. This means that the backoff language model is being too generous in allowing some sequences of unseen trigrams to be decoded.

In Figure 6 we see that trigram language model transitions tend to occur in reliable regions of decodings, while backoffs to the unigram probability of w_3 *are more likely than not* to appear in an error region, even though they occur overall only 2% of the time in non-OOV utterances.

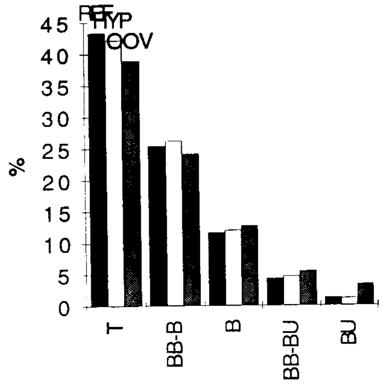

Figure 4: Overall rates of language model transition types. The left-hand bars represent rate within non-OOV reference utterances. Middle bars represent rate within the corresponding decoder output. Right-hand bars represent rate within the decoder output of OOV utterances.

These analyses motivate two modifications. One is to help the backoff language model to be more judicious in its scoring of unseen sequences of words. The section after next describes a technique called *context-based backoff capping (CBC)* that does just that. The other is to adjust the relative role of the language model and acoustic models in error-prone language model contexts. A technique for doing this is described in Section 5.

3. SEARCH

The search mechanism in Sphinx-II, described in detail in [6], is a three-pass system which is configured for these tests with a 20,000 word vocabulary. The first two passes use a backed off bigram language model to generate a word lattice with possible begin and end times. The third pass, an A^* search through the word lattice generated by the first two passes, has been extended to flexibly support long distance language models. Two such language models are referred to in this paper. The first is a backed off trigram language model and the second is the context capping (CBC) model described in Section 4. The acoustic and language model scores in Sphinx-II are combined in probability space according to the relation

$$P(w_n|h_{(1..n-1)}) = \prod_{i=1}^{n} AC(w_i) \times LM(w_i|h_{(1..i-1)})^{LW}$$

```
REF: two previous word processing leaders have SLIPPED
HYP: two previous word processing leaders have SLEPT
Error region:
                                          LM          AC          TOT
        (263,  307)    (308,  374)
REF:       SLIPPED            </s>      148324
HYP:         SLEPT            </s>                  301093       152769
        (263,  307)    (308,  374)
```

Figure 3: Hand output of ERRCMP for a simple error region. The output shows the reference transcript tokens (REF), the decoder hypothesis output tokens (HYP), and the alignment used to count WER. An error region beginning at frame 263 and ending at frame 374 has been identified. The component subregions (SLIPPED/SLEPT and </s>/</s>) of the error region and the difference in component scores within the error region are displayed. The value "148324" under "LM" is interpreted to mean that in this error region the language model score for the REF sequence was better than the language model score for the HYP sequence by 148324 score points. A similar delta value is presented for the acoustic model scores and the total combined scores. At this verbosity level segment acoustic scores, language model scores, and language model sources are not displayed.

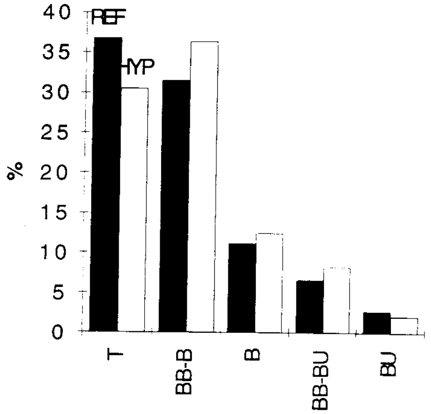

Figure 5: Rates of language model transition types within error regions.

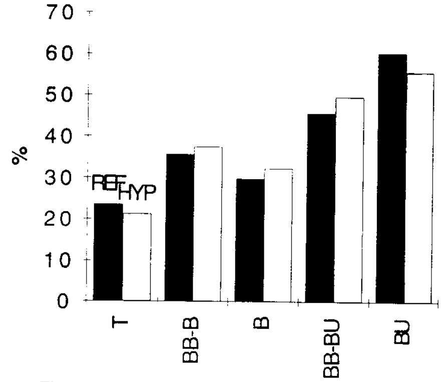

Figure 6: Relative rates of language model transition types.

where AC is the acoustic score, LM is the language model score, and LW is an optimizing parameter.

4. CORRECTING OVERESTIMATION IN THE BACKOFF MODEL

The backoff n-gram language model[4] is defined recursively as:

$$P_n(w_n|w_1^{n-1}) = \begin{cases} (1-d)C(w_1^n)\ /\ C(w_1^{n-1}) & \text{if } C(w_1^n) > 0 \\ \alpha(C(w_1^{n-1}))\cdot P_{n-1}(w_n|w_2^{n-1}) & \text{if } C(w_1^n) = 0 \end{cases} \qquad (1)$$

where d, the discount ratio, is a function of $C(w_1^n)$, and the α's are the backoff weights, calculated to satisfy the sum-to-1 probability constraints.

When $C(w_1^n) = 0$, the model assumes that the probability is proportional to the estimate provided by the n-1-gram, $P_{n-1}(w_n|w_2^{n-1})$. But for frequent n-1–grams, there may exist sufficient statistical evidence to suggest that the backed-off probabilities should in fact be much lower.

To correct this systematic overestimation, [7] proposed *Context-based Backoff Capping* (CBC)[3]: Let $C(w_1^n) = 0$. Given a global confidence level Q, to be determined empirically, calculate a confidence interval in which the true value of $P(w_n|w_1^{n-1})$ should lie, using the constraint:

$$[1 - P(w_n|w_1^{n-1})]^{C(w_1^{n-1})} \geq Q \qquad (2)$$

The confidence interval is therefore $[0 \ldots \mathrm{CAP}_Q]$, where

$$\mathrm{CAP}_Q(C(w_1^{n-1})) = (1 - Q^{1/C(w_1^{n-1})}) \qquad (3)$$

functions as a *cap* on the value of $P(w_n|w_1^{n-1})$. The backoff case of the standard model is therefore modified to:

$$P(w_n|w_1^{n-1}) = \min\left[\ \alpha(w_1^{n-1})\cdot P_{n-1}(w_n|w_2^{n-1}),\ \mathrm{CAP}_Q(C(w_1^{n-1}))\ \right] \qquad (4)$$

(See [7] for more detail.)

[3] oroginally called "Confidence Interval Capping".

Lang. Mod.	Female	Male
Baseline	160.21	184.93
Q=0.05	157.37	182.68
Q=0.1	156.80	182.20
Q=0.2	155.99	181.31
Q=0.3	155.30	180.46

Table 2: Test set perplexity on the si_dt_20 set of the 20o-nvp WSJ task, using the baseline adn the CBC model.

To test the effect of CBC, test set perplexity was measured on the reference transcripts of the test set utterances. The baseline results quoted in Table 2 are for values of Q ranging from 0.05 to 0.3. In previous perplexity based tests of the CBC (bigram) language models [7] the optimal value of Q was found to be 0.8. In the results in Table 2 it's clear that the perplexity does drop as Q is increased. However, it was clear from results quoted below with the decoder experiments that any value of Q greater than 3.0 would not yield good error rate results, so the issue of exactly how high the optimal Q value under perplexity should be was moot. The largest perplexity reduction quoted in the table is 3.1% for the female speakers and 2.4% for the males. As discussed in [8], such a small change in perplexity does not typically yield much benefit in WER.

Lang. Mod.	F OOV	F noOOV	M OOV	M noOOV
Baseline	22,0%	13.2%	24.9%	17.1%
Q=0.05	21.7%	13.1%	26.0%	18.4%
Q=0.1	21.7%	13.5%	26.0%	18.6%
Q=0.2	22.7%	14.0%	26.5%	19.2%

Table 3: Word error rate on the si_dt_20 test set broken out by utterances that contain OOV words and those that don't.

The CBC method was integrated into the third (A^*) pass of the decoder, as described in Section 3. The baseline results quoted in Table 3 are for the system configured with forward and backward passes using bigrams and the third (A^*) pass using trigrams. Values of Q=0.05, 0.1, and 0.2 are quoted as well. The only improvement found in the test set was for the female speakers with the value of Q=0.05. CBC implemented in the third pass of the decoder did not enhance the performance of the trigram language model.

5. ADJUSTING BACKED OFF CONTRIBUTIONS

Our error analysis indicated that regions in which backoff occurs are more error-prone than others. One approach to dealing with this is to modify the LW parameter value for transitions that are non-trigram, letting the language model play a different relative role with respect to the acoustic model in these cases. The decoder was modified to support a second parameter, $BOLW$, which plays a mathematically analagous role to LW but replaces LW in scoring non-trigram language model transitions. The results in Table 4 indicate that an improvement for both OOV and non-OOV utterances is provided by increasing the value of $BOLW$. Raising the value of $BOLW$ causes a heightened sensitivity to goodness or badness in a backed off language model score when compared with a trigram score. An improvement of 4% in the OOV utterances is made possible with an even stronger $BOLW$. This improvement is largely due to the fact that fewer words are used to decode each OOV word in the input data.

Backoff LW	F OOV	F noOOV
BOLW = 8	23.1%	14.3%
Base = 9.5	22.0%	13.2%
BOLW = 10	21.7%	13.0%
BOLW = 11	21.1%	13.5%
BOLW = 12	22.5%	14.2%

Table 4: Word error rate on the female half of the si_dt_20 test set, broken out by utterances that contain OOV words and those that don't. The baseline run is the one in which LW=BOLW=9.5, which was the globally optimal value for LW in isolation.

6. CONCLUSIONS and FUTURE WORK

The apparently negative experimental results quoted for the CBC scheme depend on the quality of the lattice produced under the non-capped bigram language model produced by the first two baseline passes. The lattice error rate of this system is approximately 5%. Any ommissions of correct words from the lattice made under the bigram model that might have been caught by the CBC model cannot be recovered during the A^* pass. We plan to test these ideas again with a full three-pass implementation of CBC. The results of the LW adjustment scheme are promising, and we plan experiments in which LW, $BOLW$, and related parameters are optimized jointly instead of singly.

References

1. Alleva, F., Hon, H., Huang, X., Hwang, M., Rosenfeld, R., and Weide, R. *Applying SPHINX-II to the DARPA Wall Street Journal CSR Task.* in: **DARPA Speech and Language Workshop**. Morgan Kaufmann Publishers, San Mateo, CA, 1992.
2. Chase, L. L. *A New Error Analysis Tool for Large Vocabulary Speech Recognition Systems: FORCE, ERRCMP, and ERRCNT.* Carnegie Mellon University, School of Computer Science, August 1994.
3. Alleva, F. *Personal Communication.* unpublished, 1993.
4. Katz, S. *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 (1987), pp. 400–401.
5. Paul, D. and Baker, J. *The Design for the Wall Street Journal-based CSR Corpus.* in: **DARPA Speech and Language Workshop**. Morgan Kaufmann Publishers, San Mateo, CA, 1992.
6. Alleva, F., Huang, X., and Hwang, M. *An Improved Search Algorithm for Continuous Speech Recognition.* in: **IEEE International Conference on Acoustics, Speech, and Signal Processing**. 1993.
7. Rosenfeld, R. and Huang, X. *Improvements in Stochastic Language Modeling.* in: **DARPA Speech and Language Workshop**. 1992.
8. Rosenfeld, R. *Adaptive Statistical Language Modeling: A Maximum Entropy Approach.* School of Computer Science, Carnegie Mellon University, 1994.