# ROBUST ERROR CORRECTION OF CONTINUOUS SPEECH RECOGNITION

*Eric K. Ringger & James F. Allen*
Department of Computer Science
University of Rochester
Rochester, New York 14627-0226 USA
{ringger, james}@cs.rochester.edu
http://www.cs.rochester.edu/research/trains/

## ABSTRACT

We present a post-processing technique for correcting errors committed by an arbitrary continuous speech recognizer. The technique leverages our observation that consistent recognition errors arising from mismatched training and usage conditions can be modeled and corrected. We have implemented a post-processor called SPEECHPP to correct word-level errors, and we show that this post-processing technique applies successfully when the training and usage domains differ even slightly; for the purposes of the recognizer, such a difference manifests itself as differences in the vocabulary and in the likelihoods of word collocations. We hypothesize that other differences between the training and usage conditions yield recognition errors with some consistency also. Hence, we propose that our technique be used to compensate for those mismatches as well.

## 1. INTRODUCTION

Wherever we can guarantee that the training conditions for our speech recognizer will match the conditions of usage, we can expect good recognition performance with confidence. Second, wherever we know in advance how the usage conditions will vary from the training conditions, we can design our recognizer to account for that variability. For example, we may know that previously unheard speakers will use our recognizer without much variability in accent or dialect, and we can expect good accuracy. This paper, and indeed this workshop, deals with a more challenging scenario: we would like to successfully deploy continuous speech recognition technology in settings where we have no knowledge about the usage conditions. In other words, we face the challenge of recognizing speech coming from a speaker through an unknown channel.

Possible mismatches between the training and usage conditions can occur as differences in any of the following:

- domain (vocabulary and word collocations)
- language
- speaker
- pronunciation (speaking rate, dialect, accent, etc.)
- speaker's acoustic environment
- microphone
- other properties of the channel from speaker to recognizer

For example, we have observed that a mismatch in the domain of discourse results in vocabulary mismatch and a mismatch in the frequency of collocations among vocabulary words. We have also observed that a continuous speech recognizer trained for a particular domain will commit errors with reasonable consistency whenever a speaker attempts utterances in another (even only slightly different) domain.

Our objective is to reduce speech recognition errors. We present a straightforward technique for modeling and correcting consistent recognition errors. We hypothesize that differences (other than in the domain) between the training and usage conditions yield recognition errors with some consistency also. Hence, we propose that our technique be used to compensate for those mismatches as well. We acknowledge that when the source of the error is known and the mismatch is well-understood, other techniques will most likely be superior.

We model the channel from the speaker to the output of a given recognizer as a noisy channel. We have implemented a post-processor called SPEECHPP as a Viterbi beam-search that employs language and channel models. These models are constructed with no preconceptions of the channel's nature; in this sense, the channel is unknown. For training, SPEECHPP requires only an adequate amount of human-transcribed speaker utterances gathered under the usage (*i.e.,* test) conditions and transmitted through the test channel and through the recognizer. (We have found that a few thousand words of test data are sufficient.) The channel model is derived by comparing the human transcriptions and the recognizer output in a manner similar to that used for statistical machine translation.

### 1.1. Mismatched Domain

To date, our experiments have involved a mismatch in the domain of discourse. We have used ATIS (airline travel information) data for training the recognizer and TRAINS-95 (train route planning) [1] data for testing. Here are a few examples of the kinds of errors that occur when recognizing spontaneous utterances in the TRAINS-95 domain using Sphinx-II [5] running with models trained from ATIS data. In each example, the words tagged REF indicate what was actually said, while those tagged with HYP indicate what the speech recognition (SR) system proposed. As the first example shows, many recognition errors are simple word-for-word confusions:

```
REF:  RIGHT SEND THE  TRAIN FROM MONTREAL
HYP:  RATE  SEND THAT TRAIN FROM MONTREAL
```

In the second example, a single word was replaced by more than one smaller word:

```
REF:  GO FROM CHICAGO TO TOLEDO
HYP:  GO FROM CHICAGO TO TO LEAVE
```

Perhaps surprisingly, for such a domain mismatch a simple one-for-one word substitution channel model is sufficient to yield substantial increases in word accuracy. Some further improvements in

word accuracy result from augmenting the channel model with an account of word fertility in the channel.

### 1.2. Clients and Servers

Several research labs have considered making speech recognition available on the Internet by running publicly accessible speech servers. Such servers would likely employ general-purpose language and acoustic models, but they would need to able to recognize utterances in new domains from new speakers in potentially new acoustical environments. For reduced error rates, the mismatched conditions would necessitate one of two things:

- the server itself would need to adapt and maintain new models for each speaker/connection;
- the remote client would need a way to model and correct the errors committed by the server.

For the first option, given the data (collected under the usage conditions) for training the SPEECHPP, the recognizer's models can of course be augmented, perhaps by interpolation with the new models. We will show that in the case where the recognizer's language model can be updated with data from a new domain, the post-processor trained on the *same* new data can still provide additional improvements in recognition accuracy.

For the second option, using a general-purpose SR engine makes sense because it allows a system to deal with diverse utterances from typical speakers in typical environments. If needed, the post-processor can correct the general-purpose hypothesis in a domain-specific or speaker-specific way to compensate for mismatches. Using the general-purpose system in new domains or environments requires only that the post-processor be tuned by passing a relatively small training set through the channel and recognizer for observation; the general-purpose recognizer and its models can be reused with little or no change. Note that the post-processor's training set must not necessarily be handled in batch mode; instead, the speaker could opt to supervise the training of the post-processor with word transcriptions in an online fashion. Because the post-processor is light-weight by comparison, the savings may be significant. This solution distributes the load of maintaining custom models among the clients.

This work demonstrates that a modern continuous speech recognizer can be used for robustly recognizing speech for which the recognizer was not originally trained. Furthermore, a recognizer can be used as a server with multiple unknown clients using unknown channels, as long as the clients are permitted to enroll with labeled speech.

## 2. THE MODELS AND ALGORITHM

We applied a noisy channel model and adapted techniques from statistical machine translation (*c.f.* [3]) and statistical speech recognition (*c.f.* [2, 6]) in order to model the errors that Sphinx-II makes in our domain. Briefly, the model consists of two parts: a channel model, which accounts for errors made by the SR, and the language model, which accounts for the likelihood of a sequence of words being uttered in the first place. Figure 1 illustrates the relationship of the speaker, the channel (including the SR), and the error-correcting post-processor.

More precisely, given an observed word sequence $\underline{\mathbf{w}}'$ from the SR, SPEECHPP finds the most likely original word sequence $\underline{\hat{\mathbf{w}}}$ by finding the word sequence $\underline{\mathbf{w}}$ that maximizes the expression $\mathrm{P}[\underline{\mathbf{w}}' \mid \underline{\mathbf{w}}] \cdot \mathrm{P}[\underline{\mathbf{w}}]$, where

- $\mathrm{P}[\underline{\mathbf{w}}]$ is the probability that the speaker would utter $\underline{\mathbf{w}}$,
- $\mathrm{P}[\underline{\mathbf{w}}' \mid \underline{\mathbf{w}}]$ is the probability that the SR produces the sequence $\underline{\mathbf{w}}'$ when $\underline{\mathbf{w}}$ was actually spoken.

For efficiency and due to sparse data, it is necessary to estimate these distributions with relatively simple models by making independence assumptions. For $\mathrm{P}[\underline{\mathbf{w}}]$, we train a word-bigram "back-off" language model [7, 10] from hand-transcribed dialogues previously collected with the TRAINS-95 system. For $\mathrm{P}[\underline{\mathbf{w}}' \mid \underline{\mathbf{w}}]$, we build a simple channel model that assumes independent word-for-word substitutions; *i.e.*,

$$\mathrm{P}[\underline{\mathbf{w}}' \mid \underline{\mathbf{w}}] = \prod_i \mathrm{P}[w_i' \mid w_i] \quad . \tag{1}$$

The channel model is trained by automatically aligning the hand transcriptions with the output of Sphinx-II on the utterances in the (SPEECHPP) training set and by tabulating the confusions that occurred. We say that a word is *aligned* with the word it produces.

This *one-for-one* model is insufficient for handling all SR errors, since many are the result of faulty alignment, causing *many-to-one* and *one-to-many* mappings. Accordingly, for the channel model we relax the constraint that replacement errors be aligned on a word-for-word basis. As we have seen, it is possible for a pre-channel word to "cause" multiple words or a partial word in the SR output. We will use the following utterance from the TRAINS-95 dialogues as an example.

```
REF:  TAKE A TRAIN FROM CHICAGO TO TOLEDO
HYP:  TICKET TRAIN FROM CHICAGO TO TO LEAVE
```

Following Brown *et al.*, we refer to the number of post-channel words produced by a pre-channel word in a particular alignment as the *fertility* of that pre-channel word. In the above example, "TOLEDO" is said to have a fertility of two, since it yielded two post-channel words. When a word's fertility $m$ is an integer value, it indicates that the pre-channel word resulted in $m$ post-channel words. When a word's fertility is a fraction $\frac{1}{n}$, then the word and $n-1$ neighboring words have grouped together to result in a single post-channel word. We call this situation *fractional fertility*.

We also borrow from Brown *et al.* the concept of an *alignment*, such as Figure 2. To augment our one-for-one channel model, we

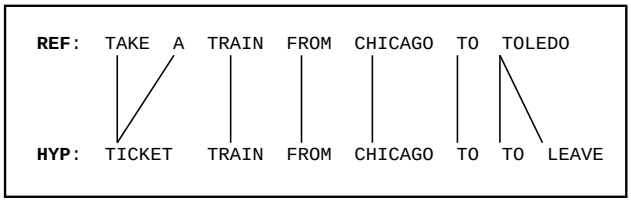


**Figure 2.** *Alignment of a Hypothesis and the Reference Transcription.*

require a probabilistic model of fertility and alignment. Our fertility model consists of several components, one for each fertility value we wish to model. For the component that models fertility two events, we have a distribution $\mathrm{P}[w_1', w_2' \mid w]$. In other words, we model the probability that pre-channel word $w$ is replaced by the two words $w_1$ and $w_2$ in the post-channel sequence. To build the fertility two model, we count the number of times that each pre-channel word $w$ is recognized as a pair $w_1', w_2'$ and compute

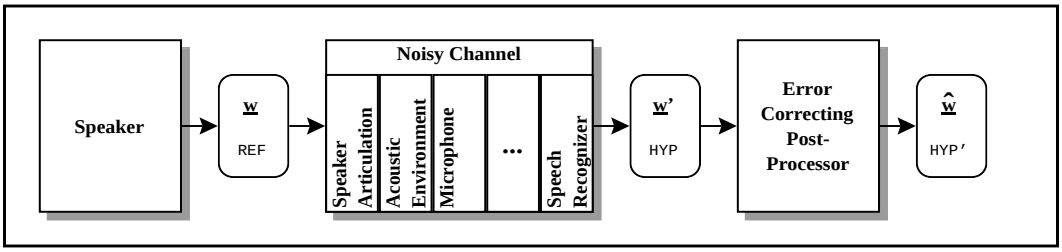


**Figure 1.** *Recovering Word-Sequences Corrupted in a Noisy Channel.*

$P[w'_1, w'_2 \mid w]$ accordingly. Similarly, for fertility one-half events, we have a distribution $P[w' \mid w_1, w_2]$.

Incorporating the fertility models, the channel probability for a given alignment $\mathcal{A}$ of a pre-channel sequence $\underline{\mathbf{w}}$ and a post-channel sequence $\underline{\mathbf{w}}'$ is the product of the probability of each piece of the alignment; *i.e.*,

$$P[\underline{\mathbf{w}}', \mathcal{A} \mid \underline{\mathbf{w}}] = \prod_{j=1}^{|\mathcal{A}|} P[\underline{\mathbf{w}}'_{\mathcal{A}.\mathtt{post}_j} \mid \underline{\mathbf{w}}_{\mathcal{A}.\mathtt{pre}_j}] \quad . \qquad (2)$$

SPEECHPP searches among possible pre-channel sequences $\underline{\mathbf{w}}$ for the most likely correction of a given post-channel sequence $\underline{\mathbf{w}}'$. The search pursues the sequence that yields the greatest value of $P[\underline{\mathbf{w}}] \cdot P[\underline{\mathbf{w}}' \mid \underline{\mathbf{w}}]$ by building possible source sequences $\underline{\mathbf{w}}$ one word at a time and scoring them. At stage $i$ of the search, each hypothesis built at stage $i-1$ is extended in all possible ways. Possible extensions are dictated by the channel model components. Given, the $i$-th post-channel word $w'_i$, if the channel model predicts a non-zero probability that a particular pre-channel word (or words) generated $w'_i$, then that pre-channel word(s) forms the tail of a new hypothesis. Thus, each word $w'_i$ in $\underline{\mathbf{w}}'$ is exploded (or collapsed with neighbors) using all possible combinations having non-zero probabilities in the model. While the source hypotheses are built, they are scored according to the language model and the channel model so that the most promising hypotheses can be pursued first. The search is efficient because it is dynamic programming on partial pre-channel sequence hypotheses, and because all partial hypotheses falling below a threshold offset (a beam) from the best current hypothesis are pruned. This is a Viterbi beam-search (*c.f.* [4, 8]).

## 3. EXPERIMENTAL RESULTS

### 3.1. Simple Channel Model

This subsection presents results based only on the one-for-one channel model and a back-off bigram language model. Having a relatively small number of TRAINS-95 dialogues for training, we wanted to investigate how well the data could be employed in models for both the SR and the SPEECHPP. We ran several experiments to weigh our options. For a baseline, we built a class-based back-off language model for Sphinx-II using only transcriptions of ATIS spoken utterances. Using this model, the performance of Sphinx-II alone was 58.7% on utterances in the TRAINS-95 domain. Note that this figure is not necessarily an indictment of Sphinx-II, but reflects the mismatch between the ATIS domain and the TRAINS-95 domain.

First, we used varying amounts of training data exclusively for building models for the SPEECHPP; this scenario would be most relevant if the SR were a black-box and we were unable to train its models. Second, we used varying amounts of the training data exclusively for augmenting the ATIS data to build new language models for Sphinx-II. Third, we combined the methods, using the training data both to extend the language models for Sphinx-II and to then train SPEECHPP on the errors committed by the newly trained SR.

The results of the first experiment are shown by the bottom curve of Figure 3, which indicates the performance of the SPEECHPP over the baseline Sphinx-II (at 58.7%). The leftmost square point comes from using approximately 25% of the available training data in the SPEECHPP models. The second and third points come from using approximately 50% and 75%, respectively, of the available training data. The curve clearly indicates that the SPEECHPP does a reasonable job of boosting our word recognition rates over baseline Sphinx-II. Also, performance improves with additional training data, up to a word error rate reduction of 14.9% (relative). We did not train with all of our available data, since the remainder was used for testing to determine the results via repeated leave-one-out cross-validation. The error bars in the figure indicate 95% confidence intervals.

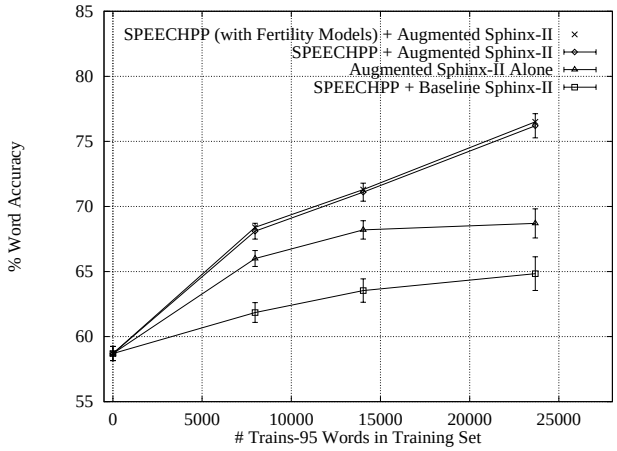


**Figure 3.** *Influence of the post-processor with additional training data.*

Similarly, the results of the second experiment are shown in the middle curve. The triangle points reflect the performance of Sphinx-II (without SPEECHPP) when using 25%, 50%, and 75% of the available training data in its LM. These results indicate that equivalent amounts of training data can be used with greater impact in the language model of the SR than in SPEECHPP.

Finally, the outcome of the third experiment is reflected in the third highest curve. Each diamond point indicates the performance of the SPEECHPP using a set of models trained on the behavior of Sphinx-II for the corresponding point from the second experiment. The results from this experiment indicate that even if the language model of the SR can be modified, then SPEECHPP trained on the same new data can still significantly improve word recognition

accuracy on a separate test set, up to a word error rate reduction of 24.0% (relative). Hence, whether the SR's models are tunable or not, SPEECHPP is in neither case redundant.

### 3.2. Fertility Channel Model

We performed additional experiments using fertility models in the channel. The results reported here are relative to those achieved by the SPEECHPP reflected in the rightmost (diamond) point of the third highest curve in the graph. Using the fertility two model along with the one-for-one model used for that reference point, we observed a 0.42% drop in substitutions, a 14.2% drop in insertions, and an 3.78% rise in deletions. As expected, the model corrects several insertion errors that were beyond the reach of the one-for-one model. However, the fertility two model is clearly not perfect, since it proposes corrections from two words to one word, causing the number of deletion errors to rise.

A second experiment involved the fertility one-half model with the one-for-one channel model. Here we have the reverse scenario from the prior experiment, as the number of deletion errors fell by 4.73%, and insertions rose by 6.78% over the base channel model. We observed a 0.93% rise in substitutions. This is also not surprising, since the model triggers search hypotheses in which one word is expanded into two, sometimes erroneously. Unfortunately, the total number of errors overall is slightly higher than without this channel model.

Using all three models together, we observed an overall increase in word accuracy of 0.32% (relative) beyond the third curve in the performance chart. This result and similar results for the other reference points in the third curve comprise the fourth and uppermost curve in the chart. Clearly, this curve falls within the confidence intervals surrounding the points of the third curve. Although the results are not statistically significant, they hold promise.

With regard to the small margins of improvement from our fertility models, we observe that the amounts of training data we have used are still small. However, the techniques are sound, and we expect that further refinements, such as smoothing (generalizing) the fertility models, will improve performance. Our current efforts focus on smoothing the fertility models using phonetic-level confusions in order to improve their contributions. Results from these experiments will be available in the near future.

## 4. CONCLUSIONS AND FUTURE WORK

We have presented a post-correction technique for overcoming speech recognition errors, based upon a noisy channel model. A recognizer using models trained for one domain does not perform well on speech in a domain even closely related to the training domain. Our experiments have shown that Sphinx-II does not perform well when moving from an air-travel information domain to a closely related train-route planning domain: as shown, it achieves less than 60% word accuracy on fluent utterances collected in problem-solving dialogues with the TRAINS-95 system. SPEECHPP can help in precisely such scenarios.

We hypothesize that this technique is more generally applicable for overcoming the problems caused by mismatches between an SR's training environment and the test environment. The only prerequisite is an opportunity to observe the effects of the channel. We plan to conduct experiments involving other kinds of training versus usage mismatch, such as speaker gender mismatch.

We have also demonstrated that with or without the ability to tune the models of the SR, we can use the SPEECHPP to boost word recognition accuracy significantly. In the TRAINS-95 system, the techniques presented here have yielded word error rate reductions as high as 24.0% (relative).

Furthermore, the post-processing approach has an advantage over lattice and N-best list rescoring approaches for reducing SR errors: using its channel model, our post-processor can introduce words that are not available in the SR module's output (*c.f.* [9]). In the near future, we plan to pursue the use of word-lattices in place of simple word sequences and expect that they will provide more useful hypotheses to compete in the post-processor's search.

## 5. ACKNOWLEDGMENTS

We thank Alex Rudnicky, Ronald Rosenfeld, and Sunil Issar at CMU for providing Sphinx-II and related tools. Thanks also go to the TRAINS research group, in particular to George Ferguson and Brad Miller. This work was supported by the U. of Rochester CS Dept. and ONR/ARPA research grant number N00014-92-J-1512.

## REFERENCES

[1] J. F. Allen, B. W. Miller, E. K. Ringger, and T. Sikorski. A robust system for natural spoken dialogue. In *Proceedings of the 1996 Annual Meeting of the Association for Computational Linguistics (ACL'96)*. ACL, June 1996.

[2] L. R. Bahl, F. Jelinek, and R. Mercer. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 5(2):179–190, March 1983.

[3] P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16(2):79–85, June 1990.

[4] J. G. E. Forney. The Viterbi Algorithm. In *Proceedings of IEEE*, volume 61, pages 266–278. IEEE, 1973.

[5] X. D. Huang, F. Alleva, H. W. Hon, M. Y. Hwang, K. F. Lee, and R. Rosenfeld. The Sphinx-II Speech Recognition System: An Overview. *Computer, Speech and Language*, 1993.

[6] F. Jelinek. Self-Organized Language Modeling for Speech Recognition. Reprinted in [11]: 450-506, 1990.

[7] S. M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, pages 400–401. IEEE, March 1987.

[8] B. Lowerre and R. Reddy. The Harpy Speech Understanding System. In *Trends in Speech Recognition*. Speech Science Publications, Apple Valley, Minnesota, 1986. Reprinted in [11]: 576-586.

[9] M. Rayner, D. Carter, V. Digalakis, and P. Price. Combining Knowledge Sources to Reorder $N$-best Speech Hypothesis Lists. In *Proceedings ARPA Human Language Technology Workshop*, pages 212–217. ARPA, March 1994.

[10] R. Rosenfeld. The CMU Statistical Language Modeling Toolkit and its use in the 1994 ARPA CSR Evaluation. In *Proceedings of the ARPA Spoken Language Systems Technology Workshop*, San Mateo, California, January 1995. ARPA, Morgan Kaufmann.

[11] A. Waibel and K.-F. Lee, editors. *Readings in Speech Recognition*. Morgan Kaufmann, San Mateo, 1990.