A FERTILITY CHANNEL MODEL FOR POST-CORRECTION OF CONTINUOUS SPEECH RECOGNITION

Eric K. Ringger & James F. Allen

Department of Computer Science; University of Rochester; Rochester, New York 14627-0226

{ringger, james}@cs.rochester.edu

http://www.cs.rochester.edu/research/trains/

ABSTRACT

We have implemented a post-processor called SPEECHPP to correct word-level errors committed by an arbitrary speech recognizer. Applying a noisy-channel model, SPEECHPP uses a Viterbi beam-search that employs language and channel models. Previous work demonstrated that a simple word-for-word channel model was sufficient to yield substantial increases in word accuracy. This paper demonstrates that some improvements in word accuracy result from augmenting the channel model with an account of word fertility in the channel. This work further demonstrates that a modern continuous speech recognizer can be used in "black-box" fashion for robustly recognizing speech for which the recognizer was not originally trained. This work also demonstrates that in the case where the recognizer can be tuned to the new task, environment, or speaker, the post-processor can also contribute to performance improvements.

1. INTRODUCTION

Consider the scenario in which a speech recognizer (SR) could be purchased as a "black-box," having a clearly specified function and well-defined input (audio signal) and output (word sequence) but otherwise providing no hooks to the user for altering or tuning internal operations. The channel from the user to the recognizer could be arbitrarily different than the channel actually modeled during the recognizer's training process. Also, the language modeled in the recognizer can be arbitrarily different than the language used by a new user, including vocabulary and collocational likelihoods. For example, several research labs have considered making speech recognition available to the research community by running publicly accessible speech servers on the Internet. Such servers would likely employ general-purpose language and acoustic models. In order to employ such a speech server to recognize utterances in a new task from a new user in a potentially new acoustical environment, one of two things would be necessary due to the modeling mismatch:

- the recognizer itself would need to adapt its models (in unsupervised mode), or
- the remote client would need some way to correct the errors committed by the server.

Our objective is to reduce speech recognition errors. SPEECHPP, our post-processor, models the channel from the speaker to the output of a given recognizer as a noisy channel. Its models are constructed with no preconceptions of the channel's nature beyond simple observations of the channel's effects on some training data. We adopt statistical techniques (some of them from statistical machine translation) for modeling that channel in order to correct some of the errors introduced there. Previous work [8] demonstrated that a simple word-for-word channel model was sufficient to yield substantial increases in word accuracy. This paper demonstrates that some improvements in word accuracy result from augmenting the channel model with an account of word fertility in the channel. The output of SPEECHPP contains fewer errors than the output of the recognizer it was trained to post-correct. This is good in and of itself, but the error reduction also makes interpretation by higher-level modules such as a parser in a speech understanding system more reliable. This work has been done as part of the TRAINS-95 and TRAINS-96 conversational planning systems, which are aimed at successfully understanding spontaneous spoken utterances in human-computer dialogue [1]. Thus, higher word recognition rates contribute to better end-to-end performance in the dialogue system. We use the Sphinx-II [4] speech recognizer in our systems, but results similar to those presented here could have been obtained with any modern SR.

Here are a few examples of the kinds of errors that occur when recognizing spontaneous utterances in the TRAINS-95 domain using Sphinx-II and its models trained from ATIS data. They are drawn from problem-solving dialogues that we have collected from users interacting with the TRAINS-95 system. 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 next 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 AT
```

Why reduce recognition errors by post-processing the SR output? Why not simply better tune the SR's language and channel models for the task, speaker, acoustic environment, etc.? First, if the SR is a general-purpose black-box (running either locally or on the other side of a network on someone else's machine), modifying the decoding algorithm to incorporate the post-processor's model might not be an 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 tune the general-purpose hypothesis in a domain-specific or user-specific way. Porting an entire system to new domains only requires tuning the post-processor by passing a relatively small training set

through the recognizer for observation; the general-purpose recognizer and its models can be reused with little or no change. Because the post-processor is light-weight by comparison, the savings may be significant.

Second, even if the SR engine's models can be updated with new domain-specific data, the post-processor trained on the *same* new data can provide additional improvements in accuracy.

Third, several human speech phenomena are poorly modeled in current continuous speech recognizers, and recognition is accordingly impaired. This provides further motivation for the placement of the SR module into our conception of a noisy channel. One poorly modeled phenomenon is assimilation of phonetic features. Most SR engines model phonemes in a context-dependent fashion (*e.g.*, see [6]), and some attempt to model cross-word co-articulation effects (*c.f.* [6] also). However, as speaking speeds vary, the SR's models may not be well suited to the affected speech signal. Such errors can be corrected by the post-processing techniques discussed here, if enough training data from fast speakers is available.

Finally, the primary advantage to the post-processing approach over existing approaches for overcoming SR errors lies in its ability to introduce options that are not available in the SR module's output. Existing rescoring tactics cannot do so (*c.f.* [7]).

2. THE MODELS AND ALGORITHM

SPEECHPP yields fewer errors by effectively refining and tuning the vocabulary and language model used by the SR. To achieve this, we applied a noisy channel model and adapted techniques from statistical machine translation (such as [3]) and statistical speech recognition (*c.f.* [2]) 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.

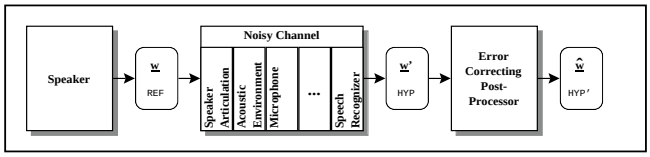

Figure 1. *Recovering Word-Sequences Corrupted in a Noisy Channel.*

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 user would utter sequence $\underline{\mathbf{w}}$,
- $\mathrm{P}[\underline{\mathbf{w}}' \mid \underline{\mathbf{w}}]$ 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 $\mathrm{P}[w]$, we train a word-bigram "back-off" language model [5] 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. For the channel model, we relax the constraint that replacement errors be aligned on a word-for-word basis, since not all recognition errors consist of simple replacement of one word by another. 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 k is an integer value, it indicates that the pre-channel word resulted in k 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 require a

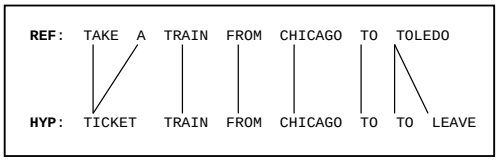

Figure 2. *Alignment of a Hypothesis and the Reference Transcription.*

probabilistic model of fertility and alignment. Initially, we thought that this model could simply consist of $\mathrm{P}[k \mid w]$ indicating how likely each word w in the pre-channel vocabulary has a particular fertility k. However, for our experiments, such a model could not be adequately constructed due to the sparseness of the training set. Instead, our fertility model consists of several components, one for each k 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. Similarly, for fertility one-half events, we have a distribution $\mathrm{P}[w' \mid w_1, w_2]$.

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 $\mathrm{P}[\underline{\mathbf{w}}] \cdot \mathrm{P}[\underline{\mathbf{w}}' \mid \underline{\mathbf{w}}]$ 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, if the channel model predicts a

non-zero probability that a particular pre-channel word (or words) generated that word, then that pre-channel word forms the tail of a new hypothesis. Thus, each word 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 from the best current hypothesis (a beam) are pruned. This is a Viterbi beam-search.

Observe that in the initial conception of the fertility model, the channel model scored only *the number of words* used to replace a particular word, and the language model scored the contents of the replacement. This was motivated by the related approach of Brown *et al.*, who appear to have taken this direction because their language model was sufficiently dense to accurately score the replacement contents. Having a relatively small amount of training data, our model is not nearly as dense as theirs, so we handle the problem in the fertility model, as described above, by tabulating only those replacements observed in the training session. For example, 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 $\mathrm{P}[w'_1, w'_2 \mid w]$.

3. EXPERIMENTAL RESULTS

3.1. Simple Channel Model

This section presents results that use only 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 models and the TRAINS-95 task.

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 model(s). Second, we used varying amounts of the training data exclusively for augmenting the ATIS data to build 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 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. The first 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.

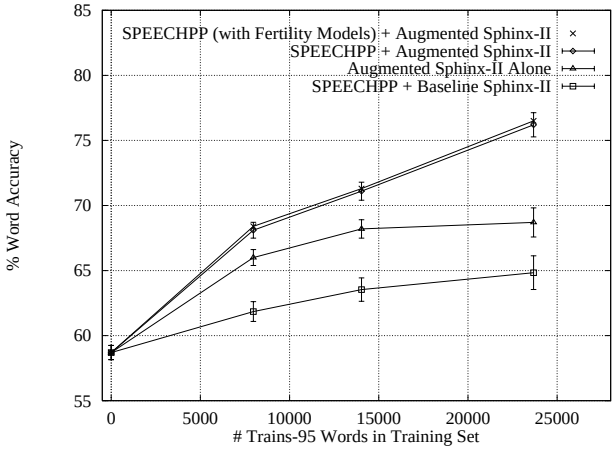

Figure 3. *Influence of the post-processor with additional training data.*

Similarly, the results of the second experiment are shown in the middle curve. The 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 uppermost curve. Each 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 point of the third 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.

3.3. Fertility Model with Silence Cues

Silence cues in the SR's hypothesis can help prevent some of the deletion errors triggered by fertility k models, for $k > 1$. We performed experiments involving the use of silence marks in the output of Sphinx-II. For example, `GO FROM CHICAGO TO TO <SIL> LEAVE FROM HERE` should not be transformed into `GO FROM CHICAGO TO TOLEDO FROM HERE` by SPEECHPP even though P[`TOLEDO | TO LEAVE`] > 0. Out of 1263 utterances (8164 words) in the test set, only three deletion errors were prevented above and beyond the fertility two results detailed above.

4. DISCUSSION

Existing continuous speech recognition techniques do not perform well when the training environment differs from the testing environment. In other words, portability is not a feature of the state of the art. For example, if the microphone (type) used to gather training data is not used to gather the testing data, or if other critical aspects of the acoustic environment change, then performance on the test set suffers dramatically. Research seeking robust acoustic features has been partially successful in remedying this particular problem. Likewise, a recognizer using models trained for one task does not perform well on speech in a task even closely related to the training task. Our experiments have shown that Sphinx-II does not perform well when moving from an air-travel reservation task to a train-route planning task: as shown, it achieves less than 60% word accuracy on fluent utterances collected in problem-solving dialogues with the TRAINS-95 system. In those experiments, the acoustic model and the class-based language model were trained on ATIS data. Similarly, a recognizer built using HTK [9] on human-human speech (Trains Dialogue Corpus) performed poorly on computer-human speech. SPEECHPP can help in precisely these scenarios.

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 largely insufficient. However, the techniques are sound, and we expect that further refinements, such as smoothing (generalizing) the fertility models, will improve performance.

5. CONCLUSIONS AND FUTURE WORK

We have presented a post-correction technique for overcoming speech recognition errors, based upon a noisy channel model. This technique is generally applicable for overcoming the problems caused by mismatches between an SR's training environment and the test environment. The only pre-requisite is sufficient test data so that the the behavior of the channel on the test environment can be sufficiently observed.

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).

We plan to further augment the fertility channel model to handle more complex cases. For example, the following (partial) utterance contains several errors, including a more complex example in which adjacent words (`WE COULD`) are misrecognized in such a way that the two hypothesized words overlap the boundary between the reference words:

```
REF:     GREAT OKAY NOW WE COULD GO FROM SAY ...
HYP: I'M GREAT OKAY NOW WEEK IT  GO FROM CITY
```

We expect that a two-for-two component (and other m-for-n components) in the channel model will handle such errors.

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 process. We also expect silence cues to play a more significant role then. We will also investigate how explicitly including silence and other simple prosodic cues in our channel models can assist in improving the SPEECHPP's hypotheses.

6. 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] 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.

[5] 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.

[6] K.-F. Lee. *Automatic Speech Recognition: the Development of the SPHINX System*. Kluwer Academic, Boston, 1989.

[7] 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.

[8] E. K. Ringger and J. F. Allen. Error Correction via a Post-Processor for Continuous Speech Recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, May 1996.

[9] S. J. Young and P. C. Woodland. *HTK: Hidden Markov Model Toolkit*. Entropic Research Lab., Washington, D.C., 1993.