A POST-PROCESSING SYSTEM TO YIELD REDUCED WORD ERROR RATES:

RECOGNIZER OUTPUT VOTING ERROR REDUCTION (ROVER)

Jonathan G. Fiscus
National Institute of Standards and Technology
Gaithersburg, MD 20899

Abstract - This paper describes a system developed at NIST to produce a composite Automatic Speech Recognition (ASR) system output when the outputs of multiple ASR systems are available, and for which, in many cases, the composite ASR output has lower error rate than any of the individual systems. The system implements a "voting" or rescoring process to reconcile differences in ASR system outputs. We refer to this system as the NIST Recognizer Output Voting Error Reduction (ROVER) system. As additional knowledge sources are added to an ASR system, (e.g., acoustic and language models), error rates are typically decreased. This paper describes a post-recognition process which models the output generated by multiple ASR systems as independent knowledge sources that can be combined and used to generate an output with reduced error rate. To accomplish this, the outputs of multiple of ASR systems are combined into a single, minimal cost word transition network (WTN) via iterative applications of dynamic programming (DP) alignments. The resulting network is searched by an automatic rescoring or "voting" process that selects an output sequence with the lowest score.

1 INTRODUCTION

The ROVER system seeks to yield reduced error rates for Automatic Speech Recognition (ASR) Technology, by exploiting differences in the nature of the errors made by multiple ASR systems. From this work, there is some evidence that there are significant differences in the nature of the errors made even by systems for which the differences in the number of word errors is not significant.

The two systems with lowest word error rate in the LVCSR 1997 Hub 5-E Benchmark Test Evaluation were BBN's and CMU-ISL's with 44.9% and 45.1% respectively[1]. The word error rate performance difference between the two systems is small, only 0.2%. As expected, both the Matched Pairs Sentence Segment Word Error (MAPSSWE) [1] and McNemar[1] statistical comparisons between the two systems indicates there is no statistically significant difference between the performance of the two systems.

When we compare the errorful segments identified by the MAPSSWE test, we find that out of 5919 errorful segments, BBN had 738 segments in which only BBN output an error, and CMU-ISL had 755 segments in which only CMU-ISL output an error. An interpretation of these statistics is that there are almost 1500 errorful segments that could potentially be

[1] The LVCSR 1997 Hub-5E Benchmark Test was administered by NIST. ASR system results were presented at the February 1997 Workshop.

corrected. What is needed is a means to compare errors and implement a rescoring process to identify the correct words. In some cases, an implementation of a simple "voting" process can identify the correct word.

2 THE "ROVER" SYSTEM DESCRIPTION

The ROVER system is implemented in two modules. First, the system outputs from two or more ASR systems are combined into a single word transition network. The network is created using a modification of the dynamic programming alignment protocol traditionally used by NIST to evaluate ASR technology. Once the network is generated, the second module evaluates each branching point using a voting scheme, which selects the best scoring word (with the highest number of votes) for the new transcription. Figure 1 shows the overall system architecture.

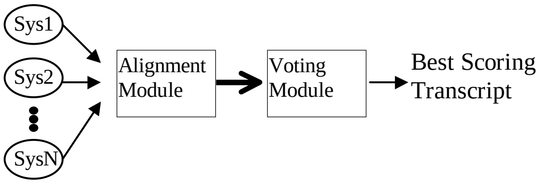


Figure 1 Rover System Architecture

2.1 MULTIPLE SEQUENCE DYNAMIC PROGRAMMING ALIGNMENT

Until now, our use of DP alignments has been applied to a pair of WTNs, (i.e. reference and hypothesis WTNs). The first stage in the ROVER system is to align the output of two or more hypothesis transcripts from ASR systems in order to generate a single, composite WTN. The second stage in the ROVER system scores the composite WTN, using any of several voting procedures.

To optimally align more than two WTNs using DP would require a hyper-dimensional search, where each dimension is an input sequence. Since such an algorithm would be difficult to implement, an approximate solution can be found using the traditional two-dimensional DP alignment process.

By taking advantage of NIST's SCLITE [2] DP alignment engine’s ability to, 1) find a minimal cost alignment between to networks, and 2) handle no-cost transition word arcs, the following procedure has been implemented to combine WTNs based on minimal cost alignments. The process can be iteratively applied for as many ASR system inputs as desired, until all inputs have been coalesced into a single composite WTN.

The first step to align and combine three or more WTNs is to create a WTN for each of the ASR system outputs. At present, the initial WTNs derived from the hypothesis files must have a linear topology, (i.e., no branching). Restricting the WTNs to linear topology simplifies the combination process. If the ROVER system is to be expanded in the future to use word lattices, this restriction will have to be overcome. Figure 2 contains three initial linear-topology WTNs that will be used to illustrate the procedure.

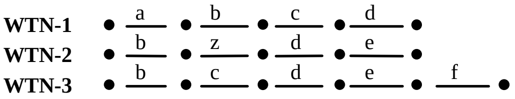


Figure 2 Initial WTNs

The first WTN is designated as the base WTN from which the composite WTN is developed. We align the second WTN to the base WTN using the DP alignment protocol and augment the base WTN with word transition arcs from the second WTN as appropriate. The alignment yields a sequence of correspondence sets between WTN-BASE and WTN-2. Figure 3 shows the 5 correspondence sets generated by the alignment between WTN-BASE and WTN-2.

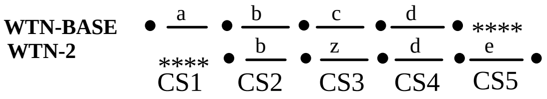


Figure 3 Aligned WTNs and correspondence set labels

Using the correspondence sets identified by the alignment process, a new, combined WTN, WTN-BASE', illustrated in Figure 4, is made by copying word transition arcs from WTN-2 into WTN-BASE. When copying arcs into WTN-BASE, the four correspondence set categories are used to determine how each arc copy is made. For a correspondence set marked as:

> Rule 1) Correct, (CS2 and CS4 in the example): a copy of the word transition arc from WTN-2 is added to the corresponding word in WTN-BASE.
>
> Rule 2) Substitution, (CS3 in the example): a copy of the word transition arc from WTN-2 is added to WTN-BASE.
>
> Rule 3) Deletion, (CS1 in the example): a no-cost, NULL word transition arc is added to WTN-BASE.
>
> Rule 4) Insertion, (CS5 in the example): a sub-WTN is created and inserted between the adjacent nodes in WTN-BASE to record the fact that the WTN-2 network supplied a word at this location. The sub-WTN is built by making a two-node WTN, that has a copy of the word transition arc from WTN-2, and *P* NULL transition arcs where *P* is the number of WTNs already previously merged into WTN-BASE. Since this is the first WTN merging in our example, P is 1.

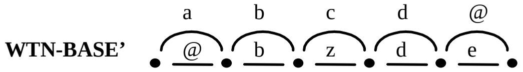


Figure 4 Composite WTN made from WTN-1 and WTN-2

Now that a new base WTN has been made, the process is repeated again to merge WTN-3 into WTN-BASE'. Figure 5 shows the final base WTN which is passed to the scoring module to select the best scoring word sequence.

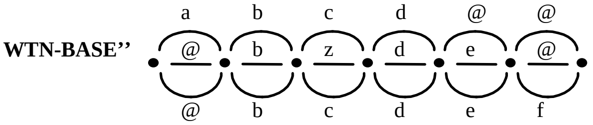


Figure 5 Final composite WTN

Iteratively combining WTNs does not guarantee an optimal composite WTN. The composite WTN is to some extent affected by the order in which the WTNs are combined. It is not the objective of the present study to investigate procedures for optimizing composite WTN generation.

2.2 WTN VOTING SEARCH MODULE

Once the composite WTN has been generated from the initial ASR system outputs, the WTN is searched by a voting or scoring module to select the best scoring word sequence.

The ROVER voting module considers each correspondence set, represented by all word transition arcs out of a graph node, as separate and independent entities on which the voting takes place. No context, forward or backward, is used to make a voting decision in the present version of ROVER, although context information (e.g., derived from N-gram language modeling) may be of significant value.

The three March 1996 LVCSR Hub-5E Benchmark Test Evaluation submissions were used as development/training sets for ROVER. The participants were required to supply with each output word, a confidence score, ranging between 0 (in effect an indication of no confidence), and 1 (in effect an indication of total confidence). The data was randomly divided into two subsets; 20% of it was designated as training data, and 80% was used for development test data. The word error rates for the three systems ranged from 44.5% to 52.0%.

We investigate three voting schemes in this paper, voting by 1) frequency of occurrence, 2) frequency of occurrence and average word confidence, and 3) frequency of occurrence and maximum confidence. We designate results generated by the voting systems as *Nist1*, *Nist2* and *Nist3* respectively.

Voting is performed by producing a set, *W(CSi)*, of unique word types within a correspondence set *CSi*. We accumulate the number of occurrences of word type *w* in correspondence set *i* in the array *N(w,i).* We divide *N(w,i)* by *Ns*, (the number of combined systems), to scale the frequency of occurrence to unity.

Similarly, depending on the voting scheme, we measure the confidence scores for word *w* in *CSi* to make the array *C(w,i).* Since NULL transition arcs do not have associated confidence scores , Confidence scores for NULL arcs are set to the trained parameter *Conf(@).*

We train a parameter, α , to be the trade off between using word frequency and confidence scores. The general scoring formula is:

$$Score(w) = \alpha(N(w,i)/Ns) + (1-\alpha)C(w,i)$$

The optimal values for both *Conf(@)* and α are found by minimizing the word error rate on a training set. This is accomplished by quantizing the parameter space into a grid of possible values, and then exhaustively searching the grid for the lowest word error rate. We call this search method a “grid-based” search.

2.2.1 Frequency of Occurrence Voting by frequency of occurrence is accomplished by setting α to 1.0, thereby ignoring all confidence scoring information.

On the development test set, the Nist1 system yields a word error rate of 43.5%, which is a 1.0% absolute reduction in word error, or 2.2% percent relative reduction.

This voting scheme has a major drawback — ties frequently occur in the word occurrence array, N(w,i). The ties were arbitrarily broken. In fact, 5320 ties out of roughly 30000 words in the development test set were broken in this manner. For this reason, alternative voting systems are attractive.

2.2.2 Average Confidence Scores. The second voting method uses confidence scores to compute an average confidence score for each word type in the array *C(w,i).* Both α and *Conf(@)* are trained a priori on the training data using the grid-search algorithm described above.

The minimum error rate on the training set was found at the parameter settings, 0.2, for α and 0.8 for *Conf(@).* These values imply that an average confidence value 0.8 is applied to all occurrences of the NULL word and that the average confidence values are four times as important as the number of occurrences of a word type.

On the development test set, the Nist2 system had a word error rate of 40.7%, which is a 3.2% absolute reduction, or 8.5% relative reduction. No score ties were arbitrarily broken on the development test set. This alone is a significant improvement over the Nist1 system.

2.2.3 MAXIMUM CONFIDENCE SCORES. The third voting method uses confidence scores to find the maximum confidence score for each word type in the array *C(w,i).* Both α and *Conf(@)* are trained a priori on the training data using grid-search algorithm described above.

The minimum error rate on the training set was found at the parameter settings, 0.7 for α , and 0.6 for *Conf(@).* These values imply that an average confidence value 0.6, somewhat less than for Nist2, is applied to all occurrences of the NULL word and that in this case, the maximum confidence values are only 0.42 (0.3/0.7) times as important as the number of occurrences of a word type. The change in alpha values from 0.2 for the Nist2 system to 0.7 for the Nist3 system is somewhat surprising, although maximum confidence values are typically larger than average confidence values.

On the development test set, the Nist3 system yields a word error rate of 40.4%, which is a 4.1% absolute reduction, or 9.2% percent relative reduction. As in the Nist2 system, no word score ties were arbitrarily broken

3.0 LVCSR 1997 HUB-5E RESULTS

The several versions of ROVER Nist1, Nist2 and Nist3, were used to post-process the LVCSR '97 Hub 5E Benchmark Test [2]submissions from BBN, CMU, CU, DRAGON and SRI with word error rates of 44.9%, 45.1%, 48.7%, 48.9% and 50.2% respectively. All trainable parameters were set based on the development set. When word error rate improvement comparisons are made between ROVER and the input systems, the lowest word error rate of 44.9% is used. Table 1 presents the error rate produced by various voting schemes implemented by ROVER

Voting Scheme	**Word Error (WE)**	**Incremental WE Reduction**	**Relative Reduction**
Nist1	39.7	5.3	11.8
Nist2	39.5	5.4	12.0
Nist3	39.4	5.6	12.5

Table 1 LVCSR Hub-5E '97 WER rates for ROVER

From Table 1, note that the word error rate for Nist1 is 39.7%. This represents an incremental word error (WE) rate reduction of 5.3%, or an 11.8% relative reduction in word error rate. When the Nist1 system is compared to each of the individual input systems, using the paired comparison statistical tests used by NIST, (the MAPSSWE, Sign, Wilcoxon and McNemar tests), all four of the tests show significant differences between the Nist1 system and the input systems with probability of greater than 99.9%.

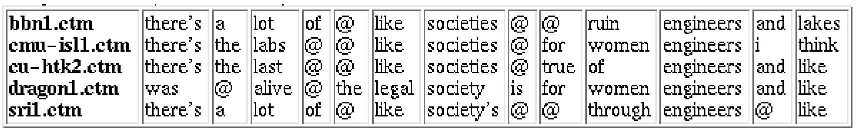
The word error rate for Nist2 is 39.5%. This represents an incremental reduction in word error rate of 5.4%, or a 12.0% relative reduction in word error rate, which was unexpectedly only slightly better than for Nist1. This improvement over Nist1 was judged to be insignificant by the MAPSSWE, Wilcoxon, Sign and McNemar statistical comparison tests. A possible explanation for the unexpected performance difference is that the parameter settings for α and *Conf(@)* on the LVCSR '96 data did not generalize to the LVCSR '97 data. We have not optimized the parameters to the test data.

[2] The LVCSR 1996 Hub-5 Benchmark Test was administered by NIST. ASR system summaries presented at the April 1996 LVCSR Workshop.

The word error rate for Nist3 is 39.4%, an incremental reduction in word error rate of 5.6%, or a 12.5% relative reduction in word error rate. The performance differences between Nist1, Nist2, and Nist3, are slight, and may by due to non-optimal parameter settings.

2.4 FURTHER ANALYSIS

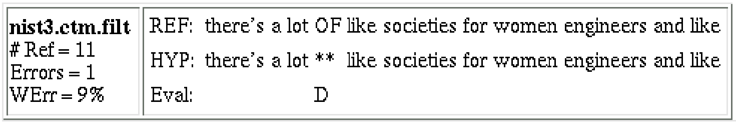
Figure 6 illustrates an output Composite WTN from aligning five ASR system outputs for one set of errorful segments from our test set. The table columns represent a correspondence set, each of which is independently considered during the voting process.



bbn1.ctm	there's	a	lot	of	@	like	societies	@	@	ruin	engineers	and	lakes
cmu-isl1.ctm	there's	the	labs	@	@	like	societies	@	for	women	engineers	i	think
cu-htk2.ctm	there's	the	last	@	@	like	societies	@	true	of	engineers	and	like
dragon1.ctm	was	@	alive	@	the	legal	society	is	for	women	engineers	and	like
sri1.ctm	there's	a	lot	of	@	like	society's	@	@	through	engineers	@	like

Figure 6 Composite WTN

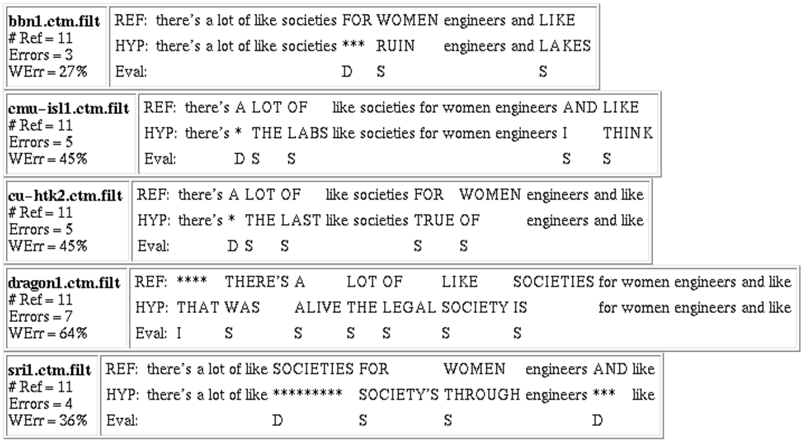
The voting process selects the best scoring word from each of the correspondence sets. Figure 7 shows an alignment of output transcript generated by Rover, aligned and scored against the reference transcript.



```
nist3.ctm.filt   REF:  there's a lot OF like societies for women engineers and like
# Ref = 11
Errors = 1       HYP:  there's a lot **  like societies for women engineers and like
WErr = 9%
                 Eval:               D
```

Figure 7 Aligned and Scored Rover Output Transcript

For this example, the Rover-generated transcript has a single error. Compared to the error rates from each for the ASR system inputs, shown in Figure 8, this is better than the lowest error rate of 3.



```
bbn1.ctm.filt     REF:  there's a lot of like societies FOR WOMEN engineers and LIKE
# Ref = 11        HYP:  there's a lot of like societies ***  RUIN     engineers and LAKES
Errors = 3        Eval:                                 D    S                      S
WErr = 27%

cmu-isl1.ctm.filt REF:  there's A LOT OF    like societies for women engineers AND LIKE
# Ref = 11        HYP:  there's *  THE LABS like societies for women engineers I    THINK
Errors = 5        Eval:         D  S   S                                       S    S
WErr = 45%

cu-htk2.ctm.filt  REF:  there's A LOT OF    like societies FOR  WOMEN engineers and like
# Ref = 11        HYP:  there's *  THE LAST like societies TRUE OF          engineers and like
Errors = 5        Eval:         D  S   S                   S    S
WErr = 45%

dragon1.ctm.filt  REF:  ****  THERE'S A     LOT OF   LIKE    SOCIETIES for women engineers and like
# Ref = 11        HYP:  THAT  WAS     ALIVE THE LEGAL SOCIETY IS       for women engineers and like
Errors = 7        Eval: I     S       S     S   S     S       S
WErr = 64%

sri1.ctm.filt     REF:  there's a lot of like SOCIETIES FOR        WOMEN   engineers AND like
# Ref = 11        HYP:  there's a lot of like ********* SOCIETY'S THROUGH engineers ***  like
Errors = 4        Eval:                       D         S         S                 D
WErr = 36%
```

Figure 8 Aligned and Scores ASR System Outputs

Of course, it is not always the case that ROVER is able to develop the best scoring hypothesis for a segment. In some cases, only one of the systems has the correct hypothesis, and its hypothesis is *out voted* by more errorful systems. Figure 9 shows the results of a scatter plot of word error rates for individual segment hypotheses, with the data ordered along the horizontal axis by increasing error rate for the Nist3 system. Note that in many cases individual systems achieve lower segment word error rate than the data for the Nist3 system (along the diagonal). Note also, however, that the centroid of the data point lies well above the diagonal, indicating that for a majority of segment points, Nist3's results have lower error rates.

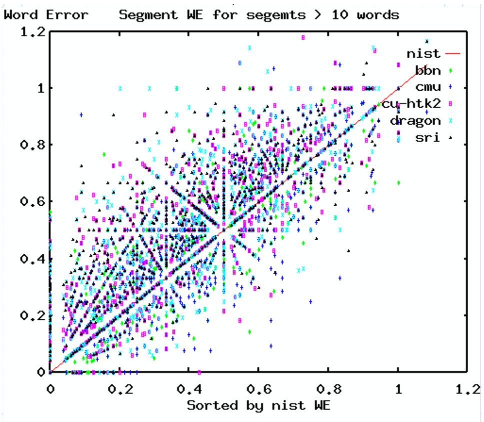


Figure 9 Scatter Plot of Word Errors

2.5 FUTURE DIRECTIONS

In planned future studies, we hope to investigate the effectiveness of incorporating other knowledge sources as input to the "voting" module. Decision Trees and artificial neural networks may be effective as alternative voting modules. We also hope to investigate alternative string alignment methods (e.g. phonologic mediation) in developing the composite WTN.

References

[1] D. Pallett, st al., "Tools for the Analysis of Benchmark Speech Recognition Tests", ICASSP 90, vol. 1, pp 97-100.

[2] The latest version of SCLITE is available from the URL "http://www.nist.gov/speech/software.htm".

Acknowledgements

Special thanks to my mentors David S. Pallett and George Doddington, without whom this paper would have not been published. Thanks to Alvin Martin who reviewed the voting scoring formulas. Of course, thanks to the ASR sites for providing NIST with their benchmark test results.