MUESLI: MULTIPLE UTTERANCE ERROR CORRECTION FOR A SPOKEN LANGUAGE INTERFACE

Federico Cesari, Horacio Franco, Gregory K. Myers, and Harry Bratt

Speech Technology and Research Laboratory

SRI International
{fico, hef, harry}@speech.sri.com, gregory.myers@sri.com

ABSTRACT

We propose a method for using all available information to help correct recognition errors in tasks that use constrained grammars of the kind used in the domain of Command and Control (CC) systems. In current spoken language CC systems, if there is a recognition error, the user repeats the same phrase multiple times until a correct recognition is achieved. This interaction can be frustrating for the user, especially at high levels of ambient noise. We aim to improve the accuracy of the error correction process by using all the previous information available at a given point, this being the previous utterances of the same input phrase and the knowledge that the previous result contained an error.

Index Terms— Error correction, command and control systems

1. INTRODUCTION

Spoken language interfaces (SLIs) are effective for applications in which the user utterances can be constrained by a known grammar. To minimize potential errors, the grammar can usually be designed so that the utterances are mutually distinguishable and the perplexity of the grammar is low. However, some required user responses, such as digit sequences, have high perplexity, and are therefore much more susceptible to errors in high-noise acoustic environments. For example, commands for soldiers who are interacting with a command and control (CC) system may include long digit sequences such as map coordinates (e.g., "Move Platoon Charlie Seven to 437224 East 172839 North"). An SLI is desirable in such applications because of its potential to improve interaction efficiency (via hands-free operation) and reduce soldiers' cognitive load. However, significant levels of noise in military environments can present a challenge to speech recognition systems. For example, while running at maximum speed (55 mph), a HMMWV produces acoustic noise at a sound pressure level (SPL) of over 100 dB inside the vehicle with the windows open. The longer the digit sequence, the more potential there is for error, especially in noisy environments. If an error is made in one of the digits, current SLI systems require the user to repeat the entire phrase, and the subsequent recognition still has the same potential for error. The user will typically repeat the same phrase multiple times until the correct result is achieved - a frustrating experience for the user. This paper describes a new method for producing a recognition result by combining information from multiple utterances to reduce the likelihood of error.

In previous work, M. Orlandi et al. [5] presented a method for correcting errors in spoken dialogue systems by dynamically adapting the language model using the error feedback from the user. In this approach the errorful 1-best hypothesis is removed from the N-best hypothesis and a new language model is constructed based on the remaining N-1 hypothesis and the M-best hypothesis of the correction. The original utterance is recognized again with this new language model.

In our approach we do something similar but the experimental setup is slightly different since the user repeats the full phrase (not just the errorful part of the utterance) on each instance (the grammar is also significantly different). The algorithm is used as a postprocessing step after we obtain the recognition output. In our algorithm we use Confusion Networks [1] generated by SRILM [2] from the recognition lattices. A new hypothesis is generated from the combination of the confusion networks for all the utterances seen for the phrase. We eliminate the 1-best hypothesis from the combination if it is the same as the 1-best hypothesis from a previous result that was errorful.

Corrections made by the user are detected by the presence of a keyword ("repeating") followed by the full utterance. This paradigm is rigid (we do not allow for a natural way of correcting a phrase like in [5]) but it becomes usable in applications in which the speakers are trained to use the system in a specific way, for example, military applications. In military applications noise is of great concern given that sometimes the user operates at a very low signal-to-noise-ratio (SNR) (0 dB or less). Using a combination of the current result and previous results for the same sentence

allows us to recover information that might be lost in one utterance but present in another instance of the same phrase.

We can approach the problem as that of combining the output of N systems where N is the number of utterances we have for each phrase. We can use an algorithm like ROVER [2] to perform the combination of the hypotheses and vote to obtain the best word sequence. In our setup there is the additional constraint that not all the phrases will be combined for each pass, given that the user will not repeat a phrase that was recognized without errors. For this reason, in the second pass we will have only a subset of the phrases from the first pass to score. Furthermore, this subset happens to be all the errorful utterances from the first pass, and hence the combination of the hypotheses in the second pass will be less effective than in a typical system combination case. The same applies for subsequent combinations. To compensate for this, we weight the hypotheses by applying a lower weight to the hypothesis previous to the current utterance in the pass. We also use confidence measures to improve the result of the confusion network combination.

2. MUESLI ALGORITHM

In the MUESLI (Multiple Utterance Error correction for Spoken Language Interface) algorithm, each spoken utterance is represented by a confusion network [1] (a sequence of word alternatives with associated probabilities). When we have several utterances of the same phrase, a new confusion network is generated that is the probabilistic combination of the confusion networks of each utterance. Confusion networks are first aligned using a Dynamic Programming algorithm that minimizes the Levenshtein distance between the topmost hypothesis in each confusion network, and then the probability distributions over words for each word slot are averaged.

The topmost hypothesis in the combined confusion network is selected as the result. This method optimally combines the acoustic evidence of the original utterance and its multiple corrections.

Confusion network weighting: We know that all recognition results previous to the last correction had at least one error. Therefore, the recognition hypothesis corresponding to the last correction has a higher prior probability of being correct than the recognition hypothesis corresponding to the previous corrections or to the original utterance. A simple heuristic to make use of this observation is to weight the confusion networks in a way that applies less probability mass to the confusion networks corresponding to the previous corrections and the original sentence and more weight to confusion network corresponding to the last correction.

Confidence: We can improve on the combination by using a measure of the confidence of the recognition output [2]. The confidence is applied to the confusion network as another weight factor. Two confidence measures were used, a confusion network confidence SC that is the geometric average of the posterior probabilities of the words in the consensus hypothesis, and a phone-based confidence measure [3] AC output by the recognizer.

Forced corrections: In previous work [5] we learned the usefulness of removing from consideration candidate hypotheses that are identical to previous recognition results that were errorful. To make use of this concept, we adapt the error-correcting method [5] to the case where the hypothesis representations are confusion networks instead of N-best lists.

For the first combination, if the topmost consensus hypothesis is equal to the previous recognition result, then we remove the topmost consensus hypothesis from the confusion network. We implement the removal of the topmost hypothesis by finding the second-best hypothesis and promoting it to the top: for every slot i in the CN we look at the difference between the posterior probabilities of the topmost word and the second most likely word d_i. We then remove the topmost word w_k from the slot k that has the smallest difference d_k, and renormalize the slot probabilities. The new topmost hypothesis of the resulting CN is called the force corrected result. For successive combinations we compare the topmost consensus hypothesis against each previous force corrected result, and, if equal, remove the topmost hypothesis as before.

Post processing: When we convert a lattice into a confusion network, the language model constraints present in the lattice structure are not preserved. For this reason there can be some hypothesis in the confusion network that are out of grammar. This can increase the sentence-level error rate of the system. To avoid this problem we perform a post processing step on the confusion network after the combination and force correction steps that eliminate all hypotheses in the network that are out of grammar. This is done by expanding all possible paths in the confusion network into an N-best list and then eliminating the hypotheses that are not accepted by the grammar that was used to recognize the utterances.

3. DATA COLLECTION

A corpus was collected at SRI that consists of 1000 phrases uttered three times each. The 1000 phrases were divided into 20 speakers, resulting in 50 unique phrases per speaker. The speakers were presented with a phrase consisting of six digits or three double digits (i.e. 34 21 56). We wanted to present the speaker with a recognition result containing real recognition errors in order to elicit behavior similar to that of a real-world application of the error correction system. To

obtain recognition errors we added noise to the waveform incrementally until at least one error was induced in the recognition output. This result was displayed to the speaker who tried to correct the error/s by repeating the phrase pre-pended with the word "repeating" to acknowledge that there was an error in the previous result. Two corrections were elicited from the speaker.

In this paradigm, the user can became frustrated quickly by not being able to correct the errors no matter how carefully that user speaks. A user who perceives that the result is going to be wrong no matter how the phrase is said will stop trying to correct the errors altogether. To avoid this situation an extra 10% of the phrases were added as decoys (that are later removed from the corpus). The decoys have the noise-mixing SNR fixed at a certain level, so that the user will get a correct recognition result at any try.

Note that the waveforms were collected in clean conditions, and the errors were induced by adding noise "on the fly" to the clean waveforms.

4. EXPERIMENTAL RESULTS

We want to assess the performance of the MUESLI algorithm at different noise levels. Errors were induced by adding increasing levels of babble noise to decrease the waveform's SNR to 10, 5, 2.5, and 0 dB. To evaluate this system we define a setup that will be representative of the interaction that a user would have with a real system. First, we evaluate the number of errors that the system produces at the different SNRs without using the MUESLI algorithm. We run a recognition pass on the initial set of utterances, which gives us the baseline performance for each SNR with no corrections. Then, we conduct two successive recognition passes to process the corrections: the sentences that contained recognition errors in the initial or previous recognition pass are *replaced* by the corresponding elicited corrections, and a new recognition pass is computed on the corrections. We call these runs C1 and C2. In a second set of experiments we conduct two successive recognition passes to process the corrections, now using the MUESLI algorithm: the sentences that contained recognition errors on the initial or previous recognition pass are *combined* with the corresponding elicited corrections by combining their corresponding confusion networks. We call these runs M1 and M2.

For each recognition pass, the error rate is recomputed over the updated set of sentences (i.e., the set of corrected sentences plus the original sentences that were correct). This process is repeated for different SNRs. This error rate in the "corrected sets" represents the residual error rate after we have applied the corrections, with or without the MUESLI algorithm. C1 and C2 are the baselines against which we compare M1 and M2, respectively. That is, we compare the benefit of combining the acoustic evidence for an utterance and its corrections, versus just using the corrections.

We also evaluated the use of different weighting of the CN associated to the original utterance and its corrections, as well as the use of confidence measures. Forced corrections based on previous user feedback were also evaluated in combination with the hypothesis combination approach.

In the following experiments we use the following notation to refer to the different variations evaluated:

1. ***Comb***: performs confusion network (CN) combination.
2. ***FC***: applies forced correction to the CN.
3. ***Comb+PConf***: uses the phonetic loop confidence measure as a weight in the CN combination.
4. ***Comb+SConf***: uses a confidence measure derived from the confusion network as a weight in the CN combination.
5. ***Comb+Alpha***: uses the weight α in the CN combination

Table 1 shows the performance of the CN combination in terms of **percent relative word error** reduction between C1 and M1 that we refer to as D1, and between C2 and M2, referred to as D2, for different SNRs. The first column shows results with the CN combination, the second column shows results using a larger weight for the CN from the last correction, and the last column shows results when weighting based on confidence is added.

SNR (dB)	***Comb***		***Comb + Alpha 0.6***		***Comb + Alpha 0.6 + PConf + SConf***	
	D1	D2	D1	D2	D1	D2
-2.5	14.0	16.4	9.1	5.9	10.7	2.9
0	12.9	15.6	10.9	9.2	11.6	4.2
2.5	4.4	16.3	6.7	7.3	9.8	4.7
5	-6.0	-9.5	6.0	0.0	7.0	0.0
10	-11.8	-10.6	1.6	-10.5	1.6	0.0

Table1: relative word error rate reduction between C1 and M1 (D1) and between C2 and M2 (D2)

We observe consistent relative word error rate reductions at low SNRs over all variants, we also observe that the use of weighting and confidence helps to improve the results of the CN combination at high SNRs.

While word error rate (WER) is a common measure of performance for ASR systems, for tasks like command and control sentence error rate (SER) represents better a sense of task completion, i.e. only correct recognized coordinates are useful. Table 2 shows the **percent relative sentence error** reduction for C1 versus M1, and for C2 versus M2, for different variations of the algorithm different SNRs.

Fig 1 shows a graph of the SER in percent for the best system variation (corresponding to the last column of Table 2). We show SERs for the first and second pass corrections, M1 and M2, for the baselines C1 and C2, and for the original sentence. This system uses CN combination with

weighting by two confidence measures and forced correction.

SNR (dB) Pass 1	*Comb*	*FC*	*Comb + FC*	*Comb +FC +Alpha 0.6*	**Comb+FC +PConf +SConf**
-2.5	2.5	0.8	6.3	5.0	**6.1**
0	-0.7	2.2	6.3	6.3	**8.3**
2.5	-8.8	6.7	14.5	11.9	**14.0**
5	-16.2	17.4	22.0	27.9	**23.2**
10	-6.4	38.7	51.6	51.6	**51.6**
Pass 2					
-2.5	-2.9	0.3	-1.0	2.7	**2.9**
0	-3.5	2.3	3.4	5.8	**6.6**
2.5	-2.2	8.3	5.4	8.8	**9.1**
5	-25.8	16.6	0.0	8.3	**7.6**
10	-17.3	27.2	0.0	10.0	**9.0**

Table 2: relative sentence error rate

From Table 2 we observe that the SER of the combinations M1 and M2 is worse than that of the corrections C1 and C2 if we only combine the utterances' CNs. In trying to improve the combination we optimized the CN weighting, but it did not produce a significant improvement in the SER unless we combined it with the forced correction operation FC. Using confidence and the alpha weight in addition to FC doesn't seem to help much in average for M1, but it does improve the SER for M2. In column 3 we can see the SER in the case of using FC directly on the corrections C1 and C2 without doing the CN combination. We observed that the SER for M1 for the systems in which we combine Comb and FC improves on the performance of FC alone. For M2, in the same cases, we observe an improvement for high SNRs.

We can observe from Table 1 and 2 that the WER is reduced by using the CN combination alone, but the SER increases, this can be attributed to the fact that CN decoding is designed to minimize WER and not SER. Using the post processing step to eliminate out of grammar hypothesis improves the SER performance but results seem to suggest that there are errors introduced by the CN combination that are in grammar and will reduce the SER. Nevertheless, the CN combination proved to be useful when combined with the forced correction process, resulting in the overall best system for the first correction and for low SNRs in the second correction.

6. CONCLUSIONS

We have proposed an approach that uses available information from previous utterances of a given sentence to help correct recognition errors, in the domain of command and control in noisy environments. The experiments showed that combining acoustic evidence from the original utterance, the correction utterances, and the user feedback, in terms of information to remove previous errorful hypotheses from the correction, was more effective than just using the correction utterances. We evaluated the "corrected set" error rate on a test set of 1000 phrases from 20 speakers and found consistent word error reductions over a wide range of SNRs.

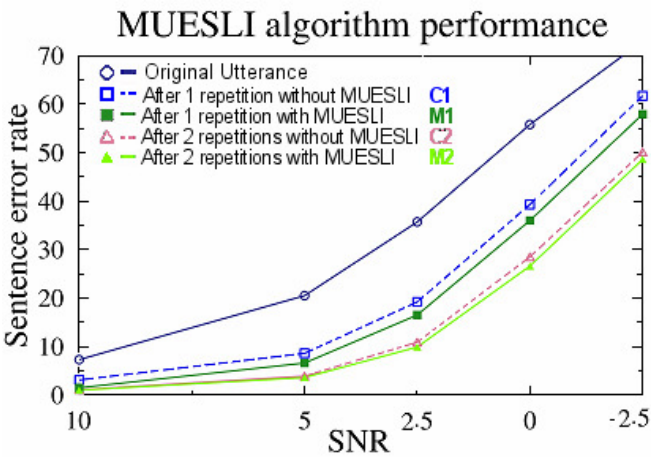

Figure 1: Sentence error rate in percent for a system that uses CN combination, confidence measures weighting, and forced correction.

REFERENCES

[1] L. Mangu, E. Brill, and A. Stolcke. "Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks". In Computer, Speech and Language, 14(4):373-400, 2000.

[2] J. Fiscus, "A post-processing system to yield reduced word error rate," in *Proc. 1997 IEEE Workshop Automatic Speech Recognition and Understanding*, 1997, pp. 347-354.

[3] Z. Rivlin, M. Cohen, V. Abrash, & T. Chung (1996), "A Phone-Dependent Confidence Measure for Utterance Rejection", *Proc. Intl. Conf. on Acoustics, Speech and Signal Processing*, 515-517, Atlanta, Georgia, 7-10 May, 1996.

[4] Franco, H., Zheng, J., Butzberger, J., Cesari, F., Frandsen, M., Arnold, J., & Gadde, V. R. R., Stolcke, A., and Abrash, V., "DynaSpeak: SRI Scalable Speech Recognizer for Embedded and Mobile Systems ". In Proc. Human Language Technology Conference, San Diego, CA, 2002.

[5] M. Orlandi, C. Culy, H Franco, "Using Dialog Corrections to Improve Speech Recognition", Proc. of the ISCA Workshop on Error Handling in Spoken Dialog Systems, Chateau-d'Oex-Vaud, Switzerland, 2003

[6] A. Stolcke. "SRILM -- an extensible language modeling toolkit". In Proc. Intl. Conf. on Spoken Language Processing, 2002.