RUN TIME INFORMATION FUSION 
IN SPEECH RECOGNITION
Chengyi Zheng1, Yonghong Yan1, 2

1Computer Science & Engineering Department, OGI School of Science & Engineering, Oregon Health & Science University, 20000 NW Walker Rd., Beaverton, OR 97006, USA
2Institute of Acoustics, Chinese Academy of Science, Beijing 100080, P.R. China

ABSTRACT
Approaches have been studied to utilize the complementary information from different knowledge sources in Automatic Speech Recognition (ASR). These approaches can be classified into two categories, the pre-recognition fusion and the post-recognition fusion. A common problem of those approaches is that complementary information is exploited either before or after recognition. To avoid unrecoverable information loss due to pruning in decoding stage, and to better utilize the complementary information, we propose a during-recognition information fusion scheme. Experimental result based on the proposed run-time fusion is reported. A significant improvement was observed.

1. INTRODUCTION
A major challenge that speech recognition research faces is the vast diversity and variability of speech, which are not just limited to noise. In order to achieve recognition robustness, we must combine efforts in signal processing (robust features), acoustic and language modeling. Most speech recognizers adopt a single "best" feature set according to the task it is facing. Also most feature representations have fixed parameters during feature extraction. Information extracted from acoustic signal can be different for features with different feature extraction methods. Human auditory studies support that there are multiple forms of signal processing in the auditory system [1]. Different responses and their outputs are combined at higher level processing [2]. In [3], it was first shown that combine information from different features and recognizers could significantly improve the machine language identification accuracy.

To utilize the complementary information from multiple features, pre-recognition approaches such as Multi-Band and Multi-Stream approaches [4, 5, 6, 7, 8] were proposed in recent years. In this approaches, feature or probability combination was performed before the actual recognition engine started. The post-recognition combination is motivated by the fact that similar complementary characteristics can be observed when using different acoustic or language models. Two systems with identical performance can have a huge difference in the errors they made. To utilize the differences in the multiple recognizers, people started to combine the outputs of several recognizers in a post-recognition combination scheme.

Approaches in this scheme include ROVER [9] and hypotheses combination [10]. All of them have demonstrated their ability to improve recognition performance.

A main problem of those approaches is that unrecoverable information loss happens during recognition cannot be avoided since no special consideration is given during decoding stage. The core of a speech recognition system is the decoder where the actual recognition is performed. During recognition, the decoder will produce a rather rich content that is largely ignored or pruned away but well worth exploring as far as fusion is concerned. We feel that it is better to perform information fusion during decoder run time: complementary information is displayed at a different stage with different form; complementary knowledge sources and processing techniques can be integrated in the same framework; a better pruning strategy can be utilized; more reliable and deep fusion is possible.

A during-recognition (run time) fusion approach is proposed in this paper. A preliminary implementation of this approach has been assessed on a large vocabulary continuous speech recognition task. The experiments demonstrate the effectiveness and potential of this approach.

2. INFORMATION FUSION REVIEW
In recent years, there’s a strong interest among ASR researchers on how to combine different features for speech recognition. The success of this research is partly due to the efficiency of improving recognition accuracy, partly due to their simplicity and easy of deployment. The existing art can be roughly classified into two categories:

2.1. Pre-Recognition Fusion
Pre-Recognition combination combines features or probabilities before conducting decoding. It can be further classified into feature combination and probability combination. Feature combination concatenates different features to form a single feature vector before acoustic modeling. The benefit of this approach is that the time dependence of different features is being exploited. Successful examples of this approach include concatenating energy and delta features with a spectral representation (such as MFCC). Probability combination is mainly used in HMM/ANN (Artificial Neural Network) hybrid systems such as Multi-Band and Multi-Stream systems [4, 5, 6, 7, 8]. A set of ANNs are
trained for each feature stream and used for probability estimation. The output of these ANNs are combined and input to a HMM decoder.

![Diagram of Probability Combination (Pre-Recognition Fusion)](image1)

Figure 1: Probability Combination (Pre-Recognition Fusion)

One advantage of probability combination is that it can be designed for parallel processing in several small models instead of a single large one. The disadvantage is that the number of ANNs needed to be trained is very large and often prohibitive for a context-dependent phone system. The drawback for both approaches is that only frame-based feature can be incorporated (or only time synchronized feature streams can be incorporated). Segmental based information, such as tones (or pitch patterns), cannot be integrated easily.

### 2.2. Post-Recognition Fusion

For the post-recognition fusion approach, the underlying mathematic assumption is the conditional independence between different feature streams during recognition of each stream. Thus decoding is performed separately on each stream independent of the decoding on the rest of the feature streams. The benefit of this approach is its simplicity and flexibility in manipulating the final recognition result. Approaches like ROVER [9] and word graph (or lattice) combination [10] all fit into this category. The time-dependency between different features is completely ignored during the recognition of each stream. There is no interaction between different features during the decoding process, and so the presence of one feature stream will not impact the course of decoding on the other features. The problem with this approach is that complementary information among different feature streams is not utilized. The mistakes made in the early decoding stage may not be recoverable at the fusion stage since the correct hypothesis may already be pruned away during decoding of each individual streams. As shown in Figure 2, recognition is performed independently on each single feature representation and the results are combined in a post recognition manner.

![Diagram of Post-Recognition Fusion](image2)

Figure 2: Post-Recognition Fusion

### 3. RUN-TIME FUSION

Current Pre- and Post-recognition undermine the possible improvement of recognition accuracy when complementary information is being utilized due to pruning error at decoder run time. The general recognition processes for the proposed work is illustrated in Figure 3. The novel part of our run-time fusion is the interaction between different feature streams during recognition (decoder run time).

One advantage of our run time information fusion is that compared with post recognition fusion, complementary information between different feature representations will be exploited during the search to avoid un-recoverable errors in post recognition processing, and the dependency (or time correlation) of different feature streams is preserved to some extent. More information can ensure that the recognizer makes fewer errors at run time by avoiding unrecoverable pruning error, and the improvement of recognition on each feature stream will contribute to the overall fused recognition performance.

![Diagram of Run-Time Fusion](image3)

Figure 3: Run-Time Fusion

Compared with a pre-recognition approach, the constraint on frame level synchronization is relaxed in our proposed work and it enables features with different time/frequency resolutions and time spans (such as segmental based features) to be readily integrated. It should be mentioned that our approach is not in opposition with the pre- or post-recognition, and they can coexist in the same system.

Under the general architecture given in Figure 3, we are investigating several approaches. Due to the length limitation of this paper, we will only introduce one of the approaches, named Constraint Fusion.

A post-recognition combination method such as ROVER lacks robustness for high error systems; the performance will be degraded when combined with some higher error systems. Since ROVER is a voting procedure, a higher error hypothesis could play a crucial role in the final output decision when there’s a tie (or near-tie) among other hypotheses. To overcome this problem, we propose a constraint fusion scheme.

A set of models will be trained independently for each feature stream. During decoding, recognition on each feature stream will run independently, except at the designated boundaries (such as phone or word boundaries). The designated boundaries are decided by decoding using only one feature
named “main feature”, and the main feature can be any feature in the total feature pool. Thus the main feature decoding sets a constraint on the search space for other features.

```
Start search on feature stream S1
```

```
Continue search

No

Synchrony point?

Yes

Conduct forced-alignment for each active path on other feature streams and fuse the likelihood

Prune the search space

search ended?

No

Yes

End
```

Figure 4: Constraint Fusion

Likelihood from other features with the same state sequence will be added (with a weight estimated under certain criteria such as Minimal Classification Error (MCE) or Maximum A Posteriori (MAP)) before the pruning. The likelihood for other feature streams can be estimated using technology similar to forced alignment (the decoded state sequence as the target of the underlying phonetic targets). The purpose is that by using additional (and hopefully complementary) information, the correct path will not be pruned away. The flow chart of this approach is given in Figure 4. This process will be conducted for every stream of feature representations and the recognition results from each stream will be fused to give the final recognition output, just as in the existing post-processing fusion methods, such as ROVER.

4. OGI SPINE2 SYSTEM

This research work is based on our Large Vocabulary Continuous Speech Recognition (LVCSR) system [11, 12]. Our LVCSR system uses decision tree based context clustering, and supports within word and cross word context-dependent phonemes (triphones). The decoder uses a two pass search strategy: the first pass generates a word graph using a simpler acoustic model (within word triphones) and language model (bigram); the second pass re-scores the word graph using a more detailed acoustic model (cross word triphone) and language model (trigram).

The test environment is the DARPA SPINE (SPEech In Noisy Environments) task. The second evaluation (SPINE2) was conducted in November 2001 [13, 14]. The test data comprises 128 speaker-environment pairs with 8 different noise environments. The test data has unseen speakers and noise types from the training data, so there will be unavoidable speaker and environment mismatch between the training and test data. Our SPINE2 system used six features; independent recognitions were conducted on each feature, and the final results were obtained by combining the outputs from these recognizers using ROVER. The six features used in our system were MFCC, TRAPS [15], Feature Net [5], TLDA [16], WMFCC [10] and SMFCC. The best official evaluation result of SPINE2 using the common language model is 38.1% WER [17]. We evaluated our approach using the same common language model.

5. EXPERIMENTAL RESULTS

We implemented an approximation of the Constraint Fusion on the SPINE2 task. The first experiment was conducted at the first pass decoding using our tree decoder, and the second set experiments were carried out during the second pass decoding by our graph decoder.

5.1. Experiment 1

In the first set of experiments, the information fusion was only performed in the phrase of graph decoding. The graph was built during first pass decoding according to one feature denoted as the “main feature”. It constrains the possible state sequences and the search space of the following search. During the second pass (graph) decoding, the search was conducted as usual but the likelihood for other features were added to the likelihood of the main feature. The weight of the addition function was

\[
Wgt (i) = 1 / N_f ,
\]

where \(N_f\) is the number of feature streams.

Although different feature streams have the same weight value during likelihood calculation, they play different roles. The search space is pre-defined by the main feature, and the other features will be involved in the decision making of the search: for example, which path could survive and which path should be pruned away before reaching the end. The motivation is to test whether the complementary information from other features could reduce search errors. Table 5 is the result when combine two features with the main feature listed in the first column and second feature on the first row of the table. The diagonal of the table shows the result of the single feature baseline system. Each row shows the result of the same main feature combined with different second features. The combined systems outperformed the single feature system in every case as shown in Table 1.

<table>
<thead>
<tr>
<th>Second</th>
<th>MFCC</th>
<th>TLDA</th>
<th>TRAPS</th>
<th>WMFCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>MFCC</td>
<td>36.7%</td>
<td>35.9%</td>
<td>36.0%</td>
<td>36.0%</td>
</tr>
<tr>
<td>TLDA</td>
<td>36.2%</td>
<td>37.4%</td>
<td>36.6%</td>
<td>36.3%</td>
</tr>
<tr>
<td>TRAPS</td>
<td>38.4%</td>
<td>38.6%</td>
<td>41.2%</td>
<td>38.4%</td>
</tr>
<tr>
<td>WMFCC</td>
<td>35.8%</td>
<td>35.7%</td>
<td>35.7%</td>
<td>36.4%</td>
</tr>
</tbody>
</table>

Table 1: WER of Constraint Fusion using two features

The fusion approach also outperformed all single feature systems under all noise categories as shown in Figure 5.
In this paper we propose an information fusion scheme to integrate different feature representations into the decoding module of a speech recognition system. Experiments showed that it outperforms a single feature system every time when fused with another feature. The simplified implementation so far has proved that information fusion during the decoding phrase not only can reduce more pruning errors but also be able to select the better path. It was rather robust in all of our experiments and outperformed the baseline system in all cases. The experimental result also shows that the complementary information our method explored is not redundant to what the ROVER does. Overall, a 7% relative word error rate reduction was observed in the first pass only decoding. This result 35.3% compared favorably to the best official evaluation result 38.1% using the common language model [17].

7. ACKNOWLEDGEMENTS

The authors thank Prof. Hynek Hermansky and his students for providing the three features (Feature Net, TLDA and TRAPS) that used in this work. Thanks also goes out to Dr. Rita Singh and Prof. Richard Stern of CMU for providing the segmentation and Language Model of SPINE2. This work is partially supported by Intel and Fonix corporations.

8. REFERENCES