*c*ROVER
Improving ROVER using Automatic Error Detection

K. Abida[1], F. Karray[1], W. Abida[2]

[1]Department of Electrical and Computer Engineering
University of Waterloo, Ontario, CANADA

[2]Voice Enabling Systems Technology Inc.
Ontario, CANADA

ICASSP 2011

Outline

1. ROVER
2. Proposed Approach
   - PMI-based Error Detection
   - Integrating Error Detection within ROVER
3. Simulation Results
   - Experimental Framework
   - Experimental Results
4. Summary and Future Directions

Motivations

- Ubiquitous use of Large Vocabulary Continuous Speech Recognition (LVCSR).
- Requirements for high accuracy, robustness and optimized speech decoders.
- Solutions:
  - Enhancing feature extraction techniques
  - Combining speech features
  - Optimizing speech decoders
  - Combining speech decoders

Decoders Output Combination

- Combine two or more speech decoders output.
  - Recognizer Output Voting Error Reduction (ROVER)
  - Confusion Network Combination (CNC)
  - Minimum Time Frame WER (fWER)
- Compile a new output that yields a reduced Word Error Rate (WER).

Outline

1. ROVER
2. Proposed Approach
   - PMI-based Error Detection
   - Integrating Error Detection within ROVER
3. Simulation Results
   - Experimental Framework
   - Experimental Results
4. Summary and Future Directions

ROVER Combination Technique

- Recognizer Output Voting Error Reduction
- Developed by NIST in 1997.
- Goal: produce a composite ASR output when the outputs of multiple ASR are available, to provide lower WER.
- Baseline technique in system combination field.
- Every new technique reports performance compared to ROVER.

ROVER Process

- Two stage process:
  - Combines multiple output into single Word Transition Network (WTN).
  - Browse the WTN by a voting process to select the output sequence.
- Three voting schemes:
  - Frequency of occurrence
  - Frequency of occurrence and Average word confidence
  - Frequency of occurrence and Maximum word confidence

ROVER Voting Schemes

- At each location in the composite WTN, a score is computed using:

$$\text{Score}(w) = \alpha\left(\frac{N(w,i)}{Ns}\right) + (1-\alpha)C(w,i)$$

  - i is the current location in the WTN
  - Ns is the total number of combined systems
  - $N(w,i)$ the frequency of word w at the position i
  - $C(w,i)$ is the confidence value for word w at the position i
  - α is set to be the trade-off between using word frequency and confidence scores. (α is optimized via a training stage)

ROVER Voting Schemes (cont'd)

- Frequency of Occurrence:
  - No word confidences used: $\alpha = 1$.
  - Frequency of occurrence used to select the winning word.
  - Problem: Ties occur very frequently and are broken randomly.
- Frequency of Occurrence and Average Word Confidence
  - Used to overcome the ties problem.
  - Overall confidence is obtained by averaging confidence of all occurrences of the same word.
- Frequency of Occurrence and Maximum Word Confidence:
  - Overall confidence is obtained by selecting the maximum confidence.

ROVER Shortcomings

- Scoring mechanism only work if the errors produced by each ASR are different from one system to the other.
- Iterative combination of word transition networks does not guarantee the optimal composite output.
- Vulnerable technique because of the use of confidence values: not reliable and much research is still underway to come up with a reliable and robust confidence measuring technique.
- Unable to use more than 1-best word sequence from each recognizer.
- Unable to outvote the erroneous ASR systems when only one single ASR output is correct.

Outline

1 ROVER

2 Proposed Approach
- PMI-based Error Detection
- Integrating Error Detection within ROVER

3 Simulation Results
- Experimental Framework
- Experimental Results

4 Summary and Future Directions

Proposed Approach

- Build the composite WTN
- Remove erroneous token from the composite WTN
- Apply ROVER voting on the new WTN

Outline

1 ROVER

2 Proposed Approach
- PMI-based Error Detection
- Integrating Error Detection within ROVER

3 Simulation Results
- Experimental Framework
- Experimental Results

4 Summary and Future Directions

PMI-based Error Detection

- Neighborhood $N(w)$
- $PMI(w_i, w_j) = log\left(\frac{P(w_i,w_j)}{P(w_i).P(w_j)}\right)$
- $P(w_i) = \frac{c(w_i)}{N}$ and $P(w_i, w_j) = \frac{c(w_i, w_j)}{N}$
- Semantic Coherence:
  - Harmonic mean: $SC(w_i) = \frac{n}{\sum_{i \neq j} \frac{1}{PMI(w_i, w_j)}}$
  - Arithmetic mean: $SC(w_i) = \frac{1}{n} \sum_{i \neq j} PMI(w_i, w_j)$
  - Maximum: $SC(w_i) = \max_{i \neq j} PMI(w_i, w_j)$
  - Sum: $SC(w_i) = \sum_{i \neq j} PMI(w_i, w_j)$.

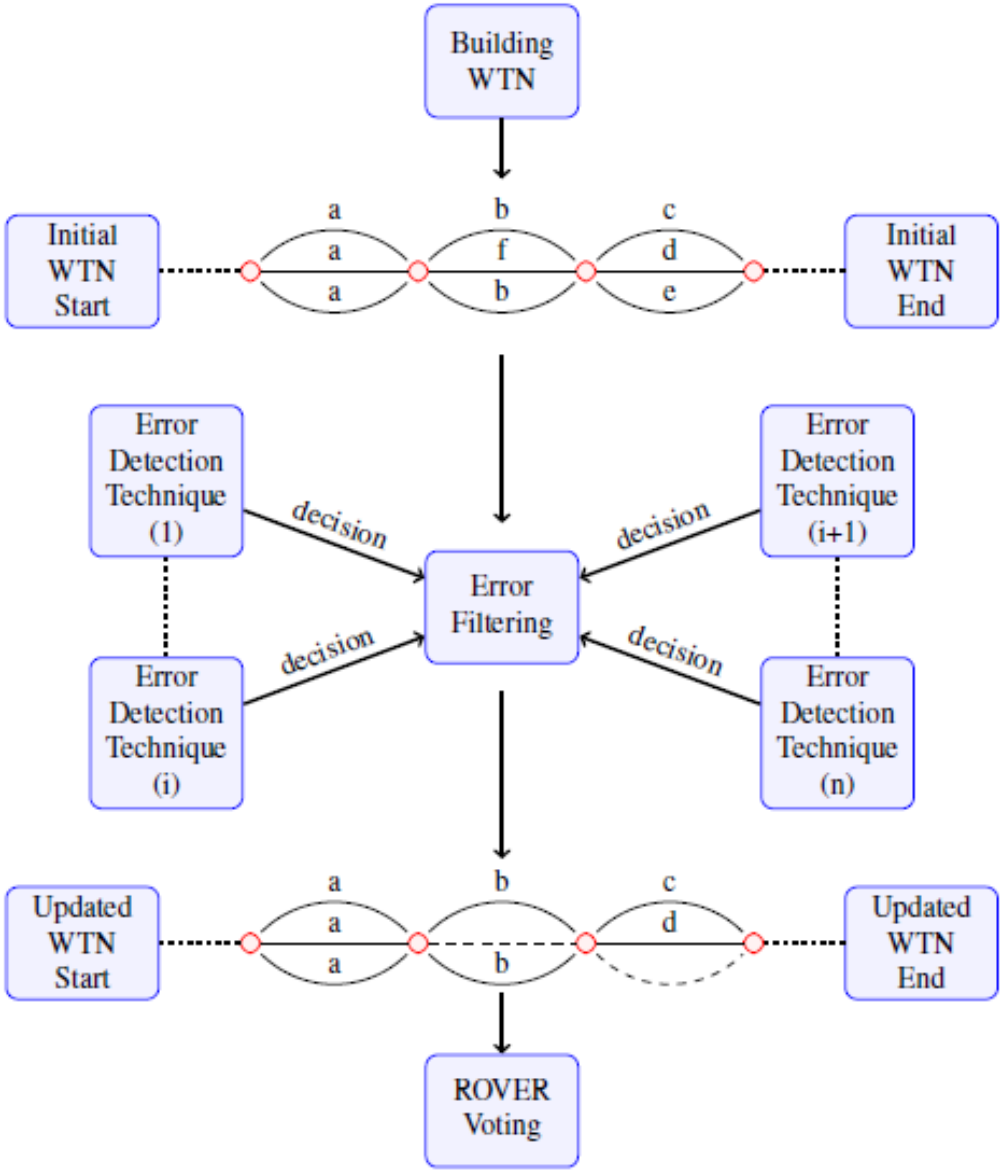
PMI-based Error Detection

Algorithm 1 PMI-based Error Detection

1: Identify the neighborhood $N(w)$.
2: Compute PMI scores $PMI(w_i, w_j)$ for all pairs of words $w_i \neq w_j$ in the neighborhood $N(w)$, including w.
3: Compute Semantic Coherence $SC(w_i)$ for every word w_i in the neighborhood $N(w)$, by aggregating the $PMI(w_i, w_j)$ scores of w_i with all $w_j \neq w_i$.
4: Define SC_{avg} to be the average of all the semantic coherence measures $SC(w_i)$ in $N(w_i)$
5: Tag the word w as an error if $SC(w) \leq K.SC_{avg}$.

Outline

1. ROVER
2. Proposed Approach
   - PMI-based Error Detection
   - Integrating Error Detection within ROVER
3. Simulation Results
   - Experimental Framework
   - Experimental Results
4. Summary and Future Directions


Building WTN
Initial WTN Start
a
a
a
b
f
b
c
d
e
Initial WTN End
Error Detection Technique (1)
Error Detection Technique (i)
Error Detection Technique (i+1)
Error Detection Technique (n)
decision
decision
decision
decision
Error Filtering
Updated WTN Start
a
a
a
b
b
c
d
Updated WTN End
ROVER Voting

Integrating Error Detection within ROVER

Algorithm 2 cROVER Procedure

1: Create the composite WTN by aligning the WTNs from the different recognizers
2: **for all** slots in the composite WTN **do**
3: Apply the PMI-based error detection
4: Remove the detected erroneous words from the slot, and replace them with the NULL transitions.
5: **end for**
6: Apply ROVER voting mechanism on the new composite WTN.

Outline

1. ROVER
2. Proposed Approach
   - PMI-based Error Detection
   - Integrating Error Detection within ROVER
3. Simulation Results
   - Experimental Framework
   - Experimental Results
4. Summary and Future Directions

Outline

1. ROVER
2. Proposed Approach
   - PMI-based Error Detection
   - Integrating Error Detection within ROVER
3. Simulation Results
   - Experimental Framework
   - Experimental Results
4. Summary and Future Directions

Simulation Framework

- Nuance V9 and Sphinx 4
- HUB4 testing framework
- Combination Settings:
  - V9 with LM98T28
  - Sphinx 4 with LM98T28
  - Sphinx 4 with LMBN99
- PMI counts: Google one trillion-token corpus
  - 13.5 million uni-grams
  - 314.8 million bi-grams

Metric Measures

- $WER = \frac{D + S + I}{N}$
- $Prec = \frac{TP}{TP + FP}$ and $Rec = \frac{TP}{TP + FN}$
- $Fmeasure = 2 * \frac{Prec * Rec}{Prec + Rec}$
- $NPV = \frac{TN}{TN + FN}$ and $PPV = \frac{TP}{TP + FP}$

Outline

1. ROVER
2. Proposed Approach
   - PMI-based Error Detection
   - Integrating Error Detection within ROVER
3. Simulation Results
   - Experimental Framework
   - Experimental Results
4. Summary and Future Directions

Error Detector Assessment

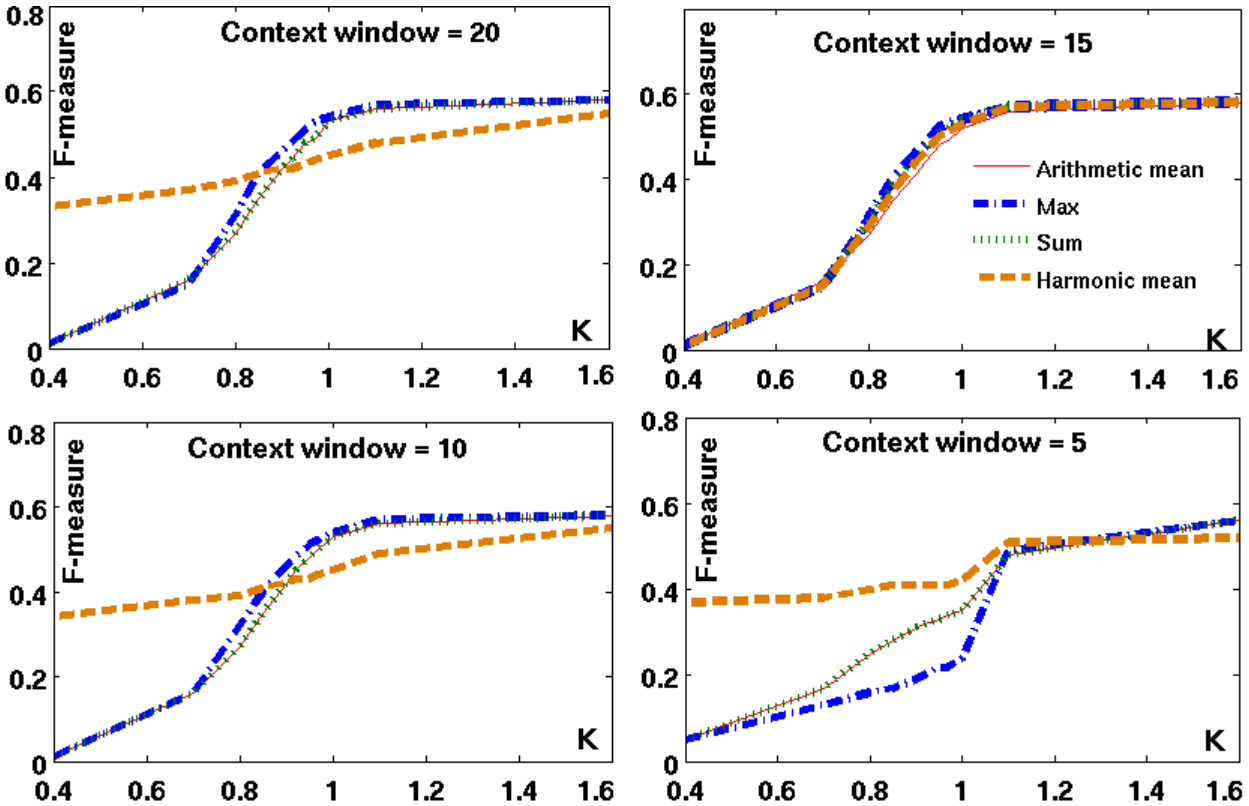


Figure: F-measure as of function of K and Aggregation methods with different Context window Sizes

Error Detector Assessment

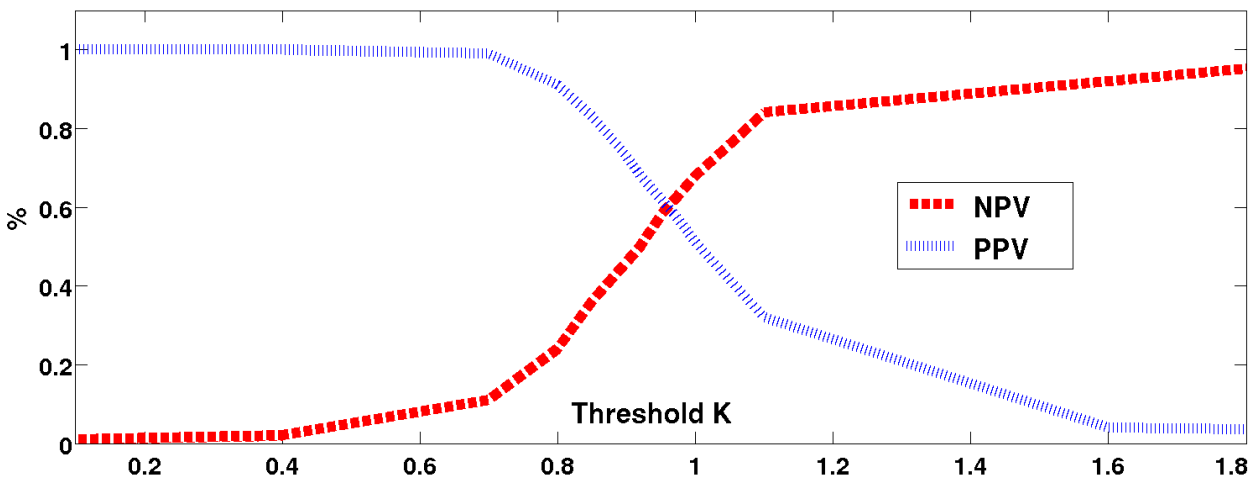


Figure: PPV and NPV Rates (Window=20, Max Aggregation)

cROVER Assessment

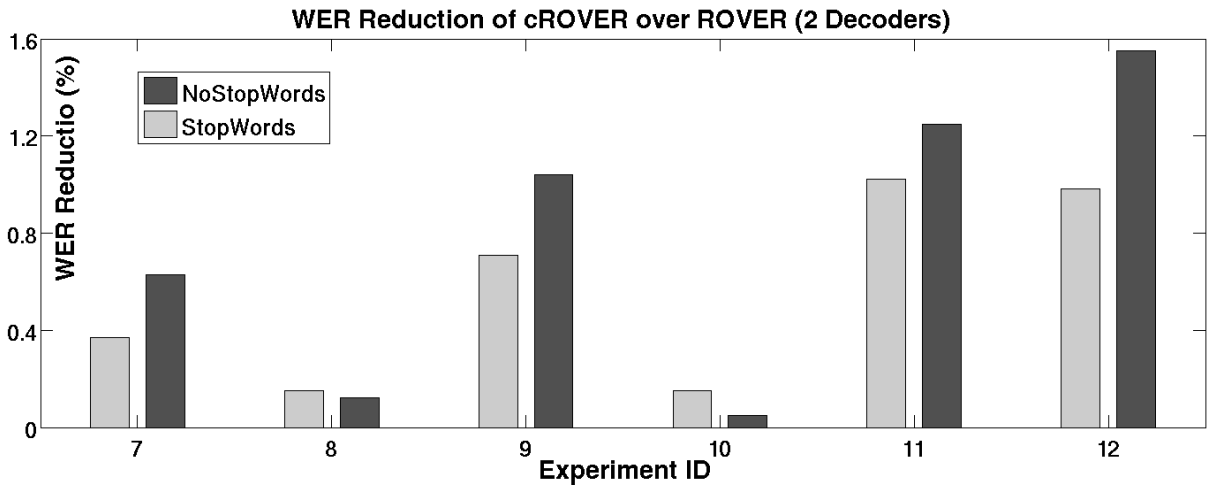


Figure: *c*ROVER vs ROVER (2 decoders)

cROVER Assessment

Figure: *c*ROVER over ROVER (3 decoders)

Outline

1. ROVER
2. Proposed Approach
   - PMI-based Error Detection
   - Integrating Error Detection within ROVER
3. Simulation Results
   - Experimental Framework
   - Experimental Results
4. Summary and Future Directions

Summary

- Novel approach to improve ROVER: cROVER
- Augmenting ROVER with contextual analysis
- PMI-based error classifier
- Up to 1.5% in WER reduction

Future Directions

- Use alternative error classifiers
- Combine error classifier
- Use alternative ASRs and Testing paradigms
- Investigate computational complexity and scalability of cROVER