Toward Using Word/Fragment Hybrid Systems

Ariya Rastrow , Abhinav Sethy, Bhuvana Ramabhadran

IBM TJ Watson Research Lab

April 17, 2009

Outline

- The simplest answer is : Recognizing OOV terms in ASR

- The simplest answer is : Recognizing OOV terms in ASR
  - All LVCSR based systems have a closed word vocabulary

- The simplest answer is : Recognizing OOV terms in ASR
  - All LVCSR based systems have a closed word vocabulary
  - The OOV term would be replaced with the closest match in the vocabulary(lack of representer)
  - Neighboring words are also often misrecognized

- The simplest answer is : Recognizing OOV terms in ASR
  - All LVCSR based systems have a closed word vocabulary
  - The OOV term would be replaced with the closest match in the vocabulary(lack of representer)
  - Neighboring words are also often misrecognized
  - Degradation of performance for later processing stages(e.g. translation,understanding, document retrieval,term detection)

- The simplest answer is : Recognizing OOV terms in ASR
  - All LVCSR based systems have a closed word vocabulary
  - The OOV term would be replaced with the closest match in the vocabulary(lack of representer)
  - Neighboring words are also often misrecognized
  - Degradation of performance for later processing stages(e.g. translation,understanding, document retrieval,term detection)
  - Although OOV rate might be low in state of the art ASR systems, *rare and unexpected events tend to be information rich*

- The simplest answer is : Recognizing OOV terms in ASR
  - All LVCSR based systems have a closed word vocabulary
  - The OOV term would be replaced with the closest match in the vocabulary(lack of representer)
  - Neighboring words are also often misrecognized
  - Degradation of performance for later processing stages(e.g. translation,understanding, document retrieval,term detection)
  - Although OOV rate might be low in state of the art ASR systems, *rare and unexpected events tend to be information rich*
- Eventually,goal in the community is to build an open vocabulary speech recognizer

- The simplest answer is : Recognizing OOV terms in ASR
  - All LVCSR based systems have a closed word vocabulary
  - The OOV term would be replaced with the closest match in the vocabulary(lack of representer)
  - Neighboring words are also often misrecognized
  - Degradation of performance for later processing stages(e.g. translation,understanding, document retrieval,term detection)
  - Although OOV rate might be low in state of the art ASR systems, *rare and unexpected events tend to be information rich*
- Eventually,goal in the community is to build an open vocabulary speech recognizer
- Fragments have the potential to provide a good trade off between coverage and accuracy

Hybrid Language Model in detail

- Step 1: N-gram pruning based fragment selection

Hybrid Language Model in detail

- Step 1: N-gram pruning based fragment selection
  - Converting LM data set(Exclude OOV) to phones, build N-gram(in our case 5-gram) phone LM and prune it(Entropy-based Pruning).
  - So, we select the set of fragments(from single phones to 5-gram phones)

Fragments → IH_N

K_L_AA_R_K

Hybrid Language Model in detail

- Step 1: N-gram pruning based fragment selection
  - Converting LM data set(Exclude OOV) to phones, build N-gram(in our case 5-gram) phone LM and prune it(Entropy-based Pruning).
  - So, we select the set of fragments(from single phones to 5-gram phones)

Fragments → IH_N

K_L_AA_R_K

- Step 2: Converting word-based LM training data into Hybrid word/fragment

Hybrid Language Model in detail

- Step 1: N-gram pruning based fragment selection
  - Converting LM data set(Exclude OOV) to phones, build N-gram(in our case 5-gram) phone LM and prune it(Entropy-based Pruning).
  - So, we select the set of fragments(from single phones to 5-gram phones)

Fragments $\longrightarrow$ IH_N

K_L_AA_R_K

- Step 2: Converting word-based LM training data into Hybrid word/fragment
  - $< s >$ THE BODY OF ZIYAD HAMDI WHO HAD BEEN SHOT WAS FOUND SOUTH OF THE CITY $< /s >$
  - $< s >$ THE BODY OF Z_IY Y_AE_D HH_AE_M D_IY WHO HAD BEEN SHOT WAS FOUND SOUTH OF THE CITY $< /s >$

Hybrid Language Model in detail

- Step 1: N-gram pruning based fragment selection
  - Converting LM data set(Exclude OOV) to phones, build N-gram(in our case 5-gram) phone LM and prune it(Entropy-based Pruning).
  - So, we select the set of fragments(from single phones to 5-gram phones)

Fragments → IH_N

K_L_AA_R_K

- Step 2: Converting word-based LM training data into Hybrid word/fragment
  - $<s>$ THE BODY OF ZIYAD HAMDI WHO HAD BEEN SHOT WAS FOUND SOUTH OF THE CITY $</s>$
  - $<s>$ THE BODY OF Z_IY Y_AE_D HH_AE_M D_IY WHO HAD BEEN SHOT WAS FOUND SOUTH OF THE CITY $</s>$
  - need to get pronunciation for OOV terms → grapheme to phone models
    ZIAD → Z IY AE D

    HAMDI→ HH AE M D IY

Hybrid Language Model in detail

- Step 1: N-gram pruning based fragment selection
  - Converting LM data set(Exclude OOV) to phones, build N-gram(in our case 5-gram) phone LM and prune it(Entropy-based Pruning).
  - So, we select the set of fragments(from single phones to 5-gram phones)

Fragments → IH_N

K_L_AA_R_K

- Step 2: Converting word-based LM training data into Hybrid word/fragment
  - $< s >$ THE BODY OF ZIYAD HAMDI WHO HAD BEEN SHOT WAS FOUND SOUTH OF THE CITY $< /s >$
  - $< s >$ THE BODY OF Z_IY Y_AE_D HH_AE_M D_IY WHO HAD BEEN SHOT WAS FOUND SOUTH OF THE CITY $< /s >$
  - need to get pronunciation for OOV terms → grapheme to phone models
    ZIAD → Z IY AE D

    HAMDI→ HH AE M D IY
  - Fragment representation of OOV is obtained by greedy search

Hybrid Language Model in detail

- Step 3: Build LM on the Hybrid word/fragment set
    - Treat fragments as individual terms
    - At this step, Hybrid LM is built and we have a LM including both words and fragments

Outline

1. Introduction

2. Hybrid Systems for OOV Detection
   - Fragment Posteriors Using Consensus
   - Additional Features
   - Evaluation
   - Experimental Setup for OOV detection
   - Results Using Various Features
   - Hybrid vs. JHU Workshop07
   - Looking At False Alarms

- The idea here is that since we have used fragments in the case of OOV for building our LM, then the existence of the fragments indicate OOV region

- The idea here is that since we have used fragments in the case of OOV for building our LM, then the existence of the fragments indicate OOV region
  - The simple case would be to search for the fragments in the decoder 1-best output
  - The better way is to search for the fragments in the lattice

- The idea here is that since we have used fragments in the case of OOV for building our LM, then the existence of the fragments indicate OOV region
  - The simple case would be to search for the fragments in the decoder 1-best output
  - The better way is to search for the fragments in the lattice
- By using fragments we would be able to not only detect OOVs but also represent them

- The idea here is that since we have used fragments in the case of OOV for building our LM, then the existence of the fragments indicate OOV region
    - The simple case would be to search for the fragments in the decoder 1-best output
    - The better way is to search for the fragments in the lattice
- By using fragments we would be able to not only detect OOVs but also represent them
    - **ASR:** TODAY TWO YOUNG GIANT PANDAS FROM CHINA ARRIVED ON A SPECIALLY R_EH_T R_OW F_IH_T IH_D FEDEX JET

- The idea here is that since we have used fragments in the case of OOV for building our LM, then the existence of the fragments indicate OOV region
  - The simple case would be to search for the fragments in the decoder 1-best output
  - The better way is to search for the fragments in the lattice
- By using fragments we would be able to not only detect OOVs but also represent them
  - **ASR:** TODAY TWO YOUNG GIANT PANDAS FROM CHINA ARRIVED ON A SPECIALLY R_EH_T R_OW F_IH_T IH_D FEDEX JET
  - **REF:** TODAY TWO YOUNG GIANT PANDAS FROM CHINA ARRIVED ON A SPECIALLY RETROFITTED FEDEX JET

Fragment Posteriors Using Consensus

Fragment Posteriors Using Consensus

Fragment Posteriors Using Consensus

- In comparison with lattices the consensuses are more compact and efficient to handle

Fragment Posteriors Using Consensus

- In comparison with lattices the consensuses are more compact and efficient to handle
- Having posterior probabilities for each hypothesis, we would be able to look not only at the existence of a fragment but also how likely that existence is.

Fragment Posteriors Using Consensus

- In comparison with lattices the consensuses are more compact and efficient to handle
- Having posterior probabilities for each hypothesis, we would be able to look not only at the existence of a fragment but also how likely that existence is.
- For any region in the confusion network we can compute an OOV score to be :

$$OOV_{score} = \sum_{f \in \{t_j\}} p(f|t_j)$$

Additional Scores

- we also explored the use of additional features that contain complimentary information such as those used in the JHU workshop. They include:

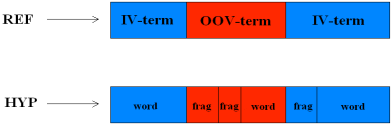
$$Word - Entropy = -\sum_{w \in \{t_j\}} p(w|t_j) \log p(w|t_j) \quad (1)$$

$$Frag - Entropy = -\sum_{f \in \{t_j\}} p(f|t_j) \log p(f|t_j) \quad (2)$$

$$LM - Score = p_{lm}(hyp_{t_j}|hyp_{t_{j-1}}) \quad (3)$$

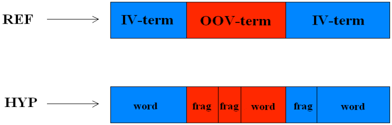
where w is a word inside region t_j and hyp_{t_j} refers to the one-best hypothesis in the current region and $hyp_{t_{j-1}}$ refers to the one-best hypothesis in the previous region. p_{lm} is the probability of seeing hyp_j given hyp_{j-1} obtained from the hybrid language model.

Evaluating OOV detection



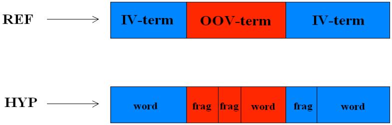
- The ASR transcript(output) is compared to the reference transcript at the *frame level*

Evaluating OOV detection



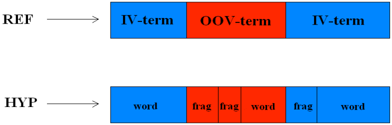
- The ASR transcript(output) is compared to the reference transcript at the *frame level*
- Each frame is assigned a score equal to the OOV score of the region it belongs to

Evaluating OOV detection



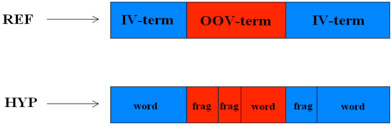
- The ASR transcript(output) is compared to the reference transcript at the *frame level*
- Each frame is assigned a score equal to the OOV score of the region it belongs to
- Each frame is tagged as belonging to an OOV or IV region.

Evaluating OOV detection



- The ASR transcript(output) is compared to the reference transcript at the *frame level*
- Each frame is assigned a score equal to the OOV score of the region it belongs to
- Each frame is tagged as belonging to an OOV or IV region.
- *Maximum Entropy*(MaxEnt) classifier is used to combine different scores.

Evaluating OOV detection



- The ASR transcript(output) is compared to the reference transcript at the *frame level*
- Each frame is assigned a score equal to the OOV score of the region it belongs to
- Each frame is tagged as belonging to an OOV or IV region.
- *Maximum Entropy*(MaxEnt) classifier is used to combine different scores.
- *False alarm* probabilities and *miss* probabilities on the set are shown in standard detection error trade-off(DET) curves

Experiment Setup for OOV detection

- Word Dictionary was limited to include 21k most frequent words(frequency greater than 5) in the acoustic training data(400 hour Hub4 Broadcast News)

Experiment Setup for OOV detection

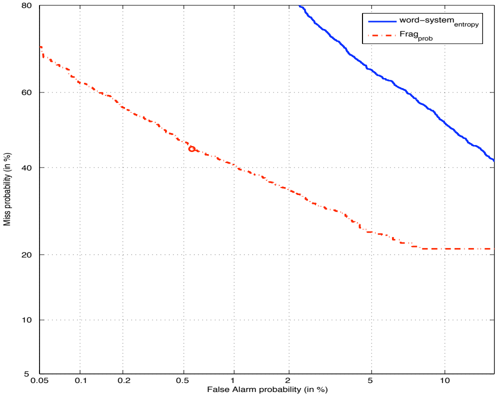
- Word Dictionary was limited to include 21k most frequent words(frequency greater than 5) in the acoustic training data(400 hour Hub4 Broadcast News)
- LM train data: 8 sources BN corpora with total 340M in-vocabulary(IV) terms and 11.6M OOV terms(OOV rate 3.4%)
    - From Hyb LM building process roughly 21k fragments were selected to be included in our hybrid dictionary

Experiment Setup for OOV detection

- Word Dictionary was limited to include 21k most frequent words(frequency greater than 5) in the acoustic training data(400 hour Hub4 Broadcast News)
- LM train data: 8 sources BN corpora with total 340M in-vocabulary(IV) terms and 11.6M OOV terms(OOV rate 3.4%)
  - From Hyb LM building process roughly 21k fragments were selected to be included in our hybrid dictionary
- Test set: RT04 BROADCAST NEWS with 4.5 hours of speech(45k words) and OOV rate of 2.8%

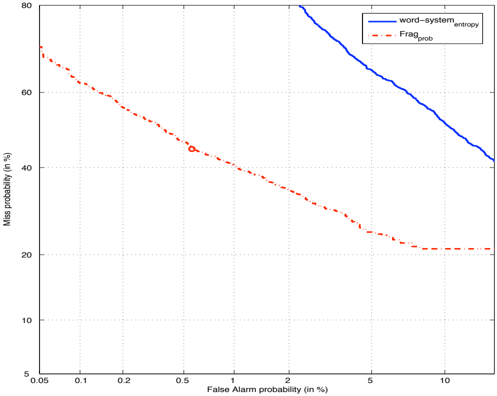
Results using various features

- Our base line is same as WS07 base line which uses Word Entropy of a word system for OOV detection



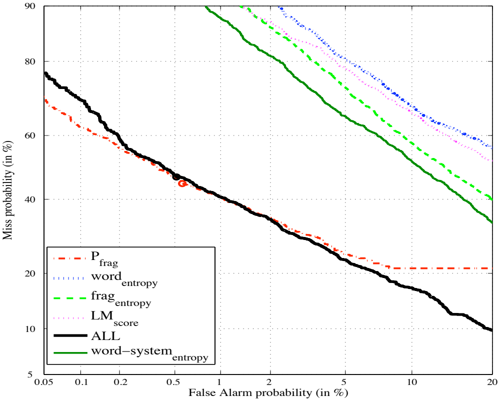
Results using various features

- Our base line is same as WS07 base line which uses Word Entropy of a word system for OOV detection
- Comparison of base line with consensus network fragment probability



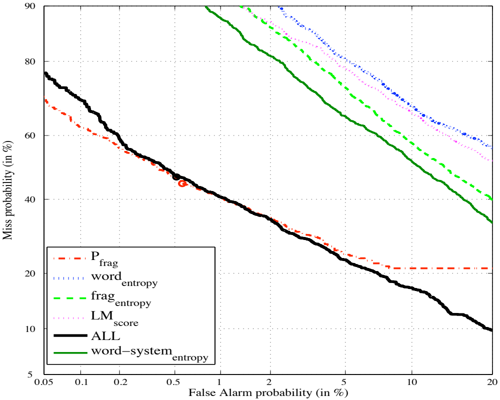
Results using various features

- Adding more feature as word entropy,fragment entropy and LM score all from hybrid system consensus network. In order to to use all features we built a MaxEnt classifier



Results using various features

- Adding more feature as word entropy,fragment entropy and LM score all from hybrid system consensus network. In order to to use all features we built a MaxEnt classifier
- Using all features we get better performance in regions with higher false alarm



Experimental Setup

- Word Dictionary was limited to include 5k most frequent words in the LM training data (defined below)

Experimental Setup

- Word Dictionary was limited to include 5k most frequent words in the LM training data (defined below)
- LM train text: WSJ LM training data (40M words, OOV rate 11.6%)
    - From Hyb LM building process roughly 24k fragments were selected to be included in our hybrid dictionary

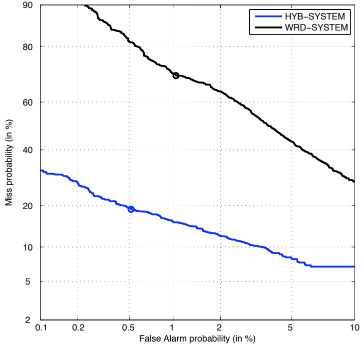
Experimental Setup

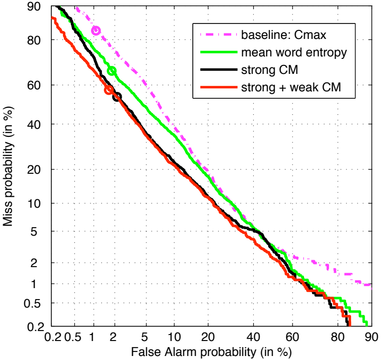
- Word Dictionary was limited to include 5k most frequent words in the LM training data (defined below)
- LM train text: WSJ LM training data (40M words, OOV rate 11.6%)
  - From Hyb LM building process roughly 24k fragments were selected to be included in our hybrid dictionary
- Test set: 1243 utterances (2.5 hours), composed from the November 1992, Hub2 5k closed test set and the WSJ1 5k open vocabulary development test set (OOV rate 5%)

Experimental Setup

- Word Dictionary was limited to include 5k most frequent words in the LM training data (defined below)
- LM train text: WSJ LM training data (40M words, OOV rate 11.6%)
    - From Hyb LM building process roughly 24k fragments were selected to be included in our hybrid dictionary
- Test set: 1243 utterances (2.5 hours), composed from the November 1992, Hub2 5k closed test set and the WSJ1 5k open vocabulary development test set (OOV rate 5%)
- In order to be consistent with workshop results, we used 8kHz down-sampled speech data

Results


HYB−SYSTEM
WRD−SYSTEM
Miss probability (in %)
False Alarm probability (in %)
90
80
60
40
20
10
5
2
0.1
0.2
0.5
1
2
5
10


baseline: Cmax
mean word entropy
strong CM
strong + weak CM
Miss probability (in %)
False Alarm probability (in %)
90
80
60
40
20
10
5
2
1
0.5
0.2
0.2
0.5
1
2
5
10
20
40
60
80
90

Looking at False Alarms

- Despite our attempt to capture OOV regions by modeling them with sub-word units, The ASR system will make errors in both OOV and IV regions
- Some examples:

GRAY $\longrightarrow$ G_R_EY 0.546 GRAY 0.275 GRADE 0.084 GREY 0.072 GREAT 0.014

ALBUM $\longrightarrow$ ALBUM 0.414 EH_L 0.257 $</s>$ 0.164 M 0.075 ELTON 0.035 YOU 0.025

Looking at False Alarms

- Despite our attempt to capture OOV regions by modeling them with sub-word units, The ASR system will make errors in both OOV and IV regions
- Some examples:

GRAY $\longrightarrow$ G_R_EY 0.546 GRAY 0.275 GRADE 0.084 GREY 0.072 GREAT 0.014

ALBUM $\longrightarrow$ ALBUM 0.414 EH_L 0.257 $< /s >$ 0.164 M 0.075 ELTON 0.035 YOU 0.025

- Is there any way to learn confusions between IV term and fragments in the consensus and try to avoid them?

Looking at False Alarms

- Despite our attempt to capture OOV regions by modeling them with sub-word units, The ASR system will make errors in both OOV and IV regions
- Some examples:

GRAY → G_R_EY 0.546 GRAY 0.275 GRADE 0.084 GREY 0.072 GREAT 0.014

ALBUM → ALBUM 0.414 EH_L 0.257 $< /s >$ 0.164 M 0.075 ELTON 0.035 YOU 0.025

- Is there any way to learn confusions between IV term and fragments in the consensus and try to avoid them?
  - Vector models to capture confusions between IV terms and fragments.

Looking at False Alarms

- We can define a vector for a given region in the test data's confusion network

$$
\begin{aligned}
\overline{V}(t_j) &= (c_1, c_2, \cdots, c_{|F|}) \\
c_i &= \frac{p(f_i | t_j)}{\sum_{f_m \in F} p(f_m | t_j)}
\end{aligned}
$$

Looking at False Alarms

- We can define a vector for a given region in the test data's confusion network

$$
\begin{aligned}
\overline{V}(t_j) &= (c_1, c_2, \cdots, c_{|F|}) \\
c_i &= \frac{p(f_i|t_j)}{\sum_{f_m \in F} p(f_m|t_j)}
\end{aligned}
$$

- Now, we define α to capture the similarity between the vector $\overline{V}(t_j)$ for the region t_j and $\overline{V}_{avg}(w)$ which is the average of $\overline{V}(w)$ over all occurences of the word w in the training data

$\alpha = p(w|t_j).sim(\overline{V}_{avg}(w), \overline{V}(t_j))$ where, w is the best hyp. word in t_j and α is the *cosine* similarity

Looking at False Alarms

- We can define a vector for a given region in the test data's confusion network

$$\begin{aligned} \overline{V}(t_j) &= (c_1, c_2, \cdots, c_{|F|}) \\ c_i &= \frac{p(f_i|t_j)}{\sum_{f_m \in F} p(f_m|t_j)} \end{aligned}$$

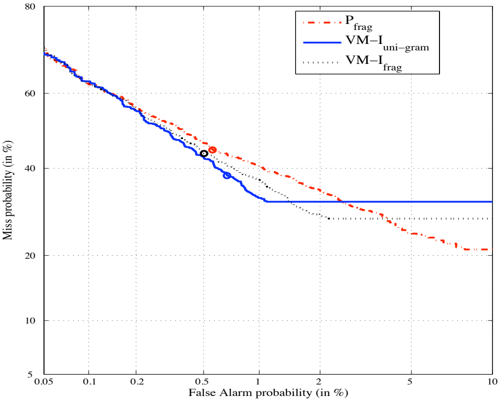
- Now, we define α to capture the similarity between the vector $\overline{V}(t_j)$ for the region t_j and $\overline{V}_{avg}(w)$ which is the average of $\overline{V}(w)$ over all occurences of the word w in the training data

$\alpha = p(w|t_j).sim(\overline{V}_{avg}(w), \overline{V}(t_j))$ where, w is the best hyp. word in t_j and α is the *cosine* similarity

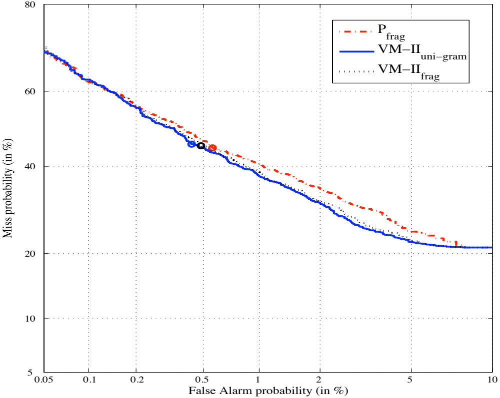
-

$$\begin{aligned} OOV_{score} &= \sum_{f \in \{t_j\}} p(f|t_j) \\ VM1 &= OOV_{score} - \alpha \\ VM2 &= OOV_{score} \cdot (1 - \alpha) \end{aligned}$$

Vector Model Results



Vector Model Results



Questions/Comments