ENSEMBLE AND RE-RANKING BASED ON LANGUAGE MODELS TO IMPROVE ASR

Shu-Fen Tsai, Shih-Chan Kuo**, Ren-Yuan Lyu***, and Jyh-Shing Roger Jang**

*Department of Computer Science and Information Engineering, National Taiwan University
**Intelligent Banking Division, E.Sun Bank
***Department of Computer Science and Information Engineering, Chang Gung University
r09922a13@ntu.edu.tw, kakushawn@gmail.com, jang@mirlab.org

ABSTRACT

We propose a strategy to improve speech recognition by selecting appropriate words to form new sentences using ensemble learning. Use traditional speech recognition methods first, and then rescore using different neural language models. Second, the decoding results of five different rescoring models were selected to select words. The choice of words is based on the importance of each word and whether the position of the word is correct. In the selection of word importance, the judgment method is majority weight and cumulative weight, and shift alignment and longest common subsequence alignment are used to determine word positions. And then the selected word representatives are reorganized to create new sentences. We compare the results of sentence ensemble and rescoring. As shown in the Aishell-1 test data, an error reduction rate of 7.30% can be achieved, verifying the effectiveness of the proposed method.

Index Terms— sentence ensemble, re-ranking, lattice rescoring, neural language models, automatic speech recognition

1. INTRODUCTION

The recognition rate is greatly improved by using neural language models for speech recognition [1] [2]. Among them, lattice rescoring is a common method of using neural language models in ASR [3] [4]. During the current decoding process, there are still some misidentified words. And different language models have their characteristics, and the decoded texts are also different. Even if the decoded text is the same, sometimes the position of the word segmentation will be inconsistent. Therefore, an ensemble learning method is proposed to improve the speech recognition rate.

The voting method is an ensemble learning model that follows the majority principle [5]. The variance is reduced through the ensemble of multiple models, thereby improving the robustness and generalization ability of the model [6] [7] [8]. Therefore, this paper combines the rescoring results of five different neural language models and performs a similar ensemble learning approach to select suitable word representatives to reorganize sentences. The five rescoring models used in this paper are RNN [9], 2-layer LSTM (2-LSTM) model with thresholds of 0.05 and 0.10 [10], and Transformer with thresholds of 0.05 and 0.10 [10].

A suitable word representative is selected from the words in the five models to form a new sentence. In the process of word selection, two factors are considered. The first factor is the importance of the word, and the second factor is based on whether the position of the word is correct. We consider two methods for evaluating the importance of words. The first method is the number of occurrences of the word. The more occurrences, the higher the probability that we think the word is the correct answer, so we choose it as the final answer. If there are two words with the same number of occurrences, the weights designed in this paper are used to select the weight, and the largest one is the final answer. This method is called majority weight (MW) in this paper. The second method is that we will accumulate the weight of the word. When the word is selected or eliminated, the weight of the word will be reset to zero, which is called cumulative weight (CW). Two methods are also devised to confirm whether the position of the word is correct. If the word that appears in the five rescoring models is the same as the result, but the word is not in the position of the result, the word is moved to that position for alignment. This article refers to this as shift alignment (SA). Another alignment method is to find the longest common subsequence (LCS) among the five models and then align the same words. This paper refers to this as LCS alignment (LA). By combining the methods of word selection and judging the correct position, four ensemble models are proposed in this paper, namely MW & SA, MW & LA, CW & SA, and CW & LA.

2. SENTENCE ENSEMBLE

In this paper, sentence ensemble is proposed as a method to improve speech recognition. First, traditional speech recognition was used, and then five neural language rescoring models were used for decoding. Using a similar ensemble learning idea, words are selected and their positions are determined,

979-8-3503-9796-3/22/$31.00

and a modified sentence is constructed after weighting each word. The model architecture is shown in Fig. 1.

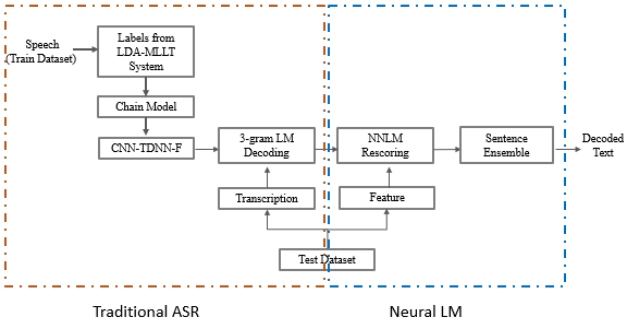

Fig. 1. Model Architecture

The weights used by the sentence ensemble method are obtained by normalizing the perplexity level (PPL) of each model-decoded sentence. Here is the formula to calculate the weights:

$$Sentence\ \ weight = \frac{\frac{1}{PPL_{S_i}}}{\sum_{i=1}^{5}\frac{1}{PPL_{S_i}}}, \tag{1}$$
$$S_i : sentence\ \ from\ \ i-th\ \ model$$

Next, the word segmentation results of each model are inconsistent. Therefore, the number of words in each sentence is different, so two alignment methods are used. Align the sentences decoded by each model. It can be divided into alignments starting from the first word, as shown in Fig. 2. Or start the alignment from the last word, as shown in Fig. 3. Then select the word representative in the same column, and use the selected word to form a new sentence as the result. Below is a detailed description of each model process.

Model	1	2	3	4	5	6	7	8	9
RNN	推进	省	直接	管理	显示	财政	体制	改革	
2-LSTM ε = 0.10	推进	上	直接	管理	县市	财政	体制	改革	
Transformer ε = 0.10	推进	省	直接	管理	县	市	财政	体制	改革
2-LSTM ε = 0.05	推进	上	直接	管理	县市	财政	体制	改革	
Transformer ε = 0.05	推进	省	直接	管理	县	市	财政	体制	改革

Fig. 2. Alignment 1

Model	1	2	3	4	5	6	7	8	9
RNN		推进	省	直接	管理	显示	财政	体制	改革
2-LSTM ε = 0.10		推进	上	直接	管理	县市	财政	体制	改革
Transformer ε = 0.10	推进	省	直接	管理	县	市	财政	体制	改革
2-LSTM ε = 0.05		推进	上	直接	管理	县市	财政	体制	改革
Transformer ε = 0.05	推进	省	直接	管理	县	市	财政	体制	改革

Fig. 3. Alignment 2

2.1. MW & SA

MW & SA is to perform shift alignment after selecting each word according to the majority weight. The practice of shift alignment is to first judge the relationship between each column of words and the result. Depending on the relationship, the following methods are used:

1. The word has been aligned with the result, clean up all words before this word.
2. The same word in this row is not in the column of result, move the word to the column of result.
3. This row does not have the same word as the result, keep the word in this column.

The pseudocode for this method is in Algorithm 1.

Algorithm 1 MW & SA

Require: A queue of sentences; each sentence comes from a different language model;
Ensure: A queue of voting results;

```
def process_sentence(sentences) :
    Queue voters = [ ]
    Queue result = [ ]
for sentence in sentences do
    voter = Voter(sentence)
    voters.enqueue(voter)
end for
while (True) :
Queue current_votes = [ ]
Dict weights = [ ]
for voter in voters do
    unmatch_votes, current_vote = voter.do_vote( )
current_votes.enqueue(current_vote)
    for vote in unmatch_votes and current_vote do
weights[vote] += voter.weight
    end for
end for
winner = find_result(current_votes, weights)
for voter in voters do
    voter.do_align(winner)
end for
if winner is \n then
    break
end if
result.enqueue(winner)
return result
```

2.2. CW & SA

In this method, the words appearing in the word selection column are used as options, and the cumulative calculation of

the word weight ($W_{i,j}$) is performed. After comparing the weights of the options, choose the one with the largest weight as the final result. The formula for the sum of the weights is as follows:

$$\begin{aligned} &W_{m,j} \longleftarrow W_{m,j} + W_{n,j} + W_{n,j-k} \\ &a_{m,j} = a_{n,j} = a_{n,j-k} \\ &0 \leq m \leq 5 \ \ and \ \ 0 \leq n \leq m \ \ and \ \ 0 \leq k \leq j \end{aligned} \tag{2}$$

After the word is selected, it will perform shift alignment. After the models are aligned, the cumulative weights of previously unselected words are updated to 0.

2.3. MW & LA

The methods mentioned above all select words and align at the same time. Therefore, it is better to speculate whether it is better to align the words first and then choose the words. Therefore, LCS alignment methods are proposed for comparison. Align words with the same common subsequence of the five models, and then select words for other words that are not in the longest common subsequence. The pseudocode for this method is in Algorithm 2.

Algorithm 2 MW & LA

Require: A queue of sentences; each sentence comes from a different language model;
Ensure: A queue of votingresultst;

```
def process_sentence(sentences) :
    Queue voters = [ ]
    Queue result = [ ]
if lcs_voting is True then
    lcs_str = lcs(sentences)
    if thenlcs_str is not empty
        Queue left_sentences = [ ]
        Queue right_sentences = [ ]
        for sentence in sentences do
            left_sentences =
            find_left_substring(sentence, lcs_str)
            right_sentences =
            find_right_substring(sentence, lcs_str)
        end for
        result += process_sentence(left_sentences)
        result.enqueue(lcs_str)
        result += process_sentence(right_sentences)
        return result
    end if
end if
```

Taking the audio file of the test data BAC009S0908W0214 as an example. To draw the word lattice according to the LCS alignment method, the same words in the five models are first merged into one state. The other words are arranged in sequence. The result of the word graph is shown in Fig. 4. From the beginning to the end of the path, when the path has multiple choices, the majority weight is used to select the appropriate word.

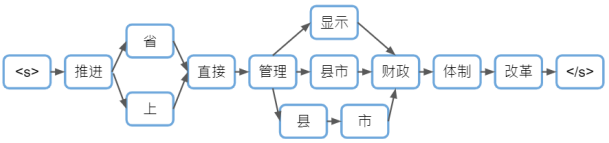

Fig. 4. LCS

2.4. CW & LA

The method is the same as MW & LA, except that the word selection is changed to cumulative weight. Therefore, first, align the same common subsequence of the five models, and then perform cumulative weight word selection on words that are not in the longest common subsequence.

3. EXPERIMENTS

3.1. Datasets and Setups

We conduct experiments on Aishell-1 [11], which includes about 178 hours of Chinese speech database. Acoustic model training and decoding using Kaldi. We trained an RNN with 1024 hidden dimensions, a 2-layer LSTM with 650 hidden dimensions, and a re-scoring model with a 6-layer Transformer with 8 heads and 512 hidden dimensions [10]. Compare with the ensemble model proposed in this paper.

3.2. Effect of Sentence Ensemble

3.2.1. Preprocessing of Sentence Ensemble

The word segmentation results are not necessarily the same for each model, resulting in some words being repeated or omitted during voting. Therefore, some modifications were made to the model. If the result of this selection is split into two words in the unselected model, the separated words will be merged. If the result of this selection is a combined word in an unselected model, the combined word will be disconnected.

Take the audio file of the test data BAC009S0908W0214 as an example. When the sentence ensemble was performed directly, there were missing words in the results due to the different word segmentation of the five models. But after preprocessing, the voting results can fill in the original missing words. Such as Fig. 5 and Fig. 6.

Weight	Model	1	2	3	4	5	6	7	8	9
0.2056	RNN	推进	省	直接	管理	显示	财政	体制	改革	
0.1895	2-LSTM ε = 0.10	推进	上	直接	管理	县市	财政	体制	改革	
0.2077	Transformer ε = 0.10	推进	省	直接	管理	县	市	财政	体制	改革
0.1895	2-LSTM ε = 0.05	推进	上	直接	管理	县市	财政	体制	改革	
0.2077	Transformer ε = 0.05	推进	省	直接	管理	县	市	财政	体制	改革
	Result	推进	省	直接	管理	县	财政	体制	改革	

Fig. 5. Raw Result

Weight	Model	1	2	3	4	5	6	7	8	9
0.2056	RNN	推进	省	直接	管理	显示	财政	体制	改革	
0.1895	2-LSTM ε = 0.10	推进	上	直接	管理	县	市	财政	体制	改革
0.2077	Transformer ε = 0.10	推进	省	直接	管理	县	市	财政	体制	改革
0.1895	2-LSTM ε = 0.05	推进	上	直接	管理	县	市	财政	体制	改革
0.2077	Transformer ε = 0.05	推进	省	直接	管理	县	市	财政	体制	改革
	Result	推进	省	直接	管理	县	市	财政	体制	改革

Fig. 6. After Preprocessing

3.2.2. Results of Sentence Ensemble

First, the CER of the neural language model rescoring is calculated, as shown in Table 1.

Table 1. CER of Rescoring

Model	CER
Chain	7.37 %
RNN	6.25 %
2-LSTM $\epsilon = 0.10$	**6.03 %**
Transformer $\epsilon = 0.10$	6.51 %
2-LSTM $\epsilon = 0.05$	6.05 %
Transformer $\epsilon = 0.05$	6.46 %

Using the four-sentence ensemble models proposed in this paper, search backward from the first word, i.e. forward, or search forward from the last word, i.e. backward. The forward search and the backward search will result in separate results. To choose a more appropriate answer, the two sentences will be better considered. If the forward and backward character counts are different, choose a character count that is closer to the character count found in most models. If the number of characters forward and backward is the same, the sentence with less PPL is selected as the final selection. This method is called SelectOne in this paper. The experimental results are shown in Table 2.

Based on the best-performing rescoring model, the 2-LSTM with a threshold of 0.10, its CER is 6.03%. It can be observed from Table 2 that the methods proposed in this paper can perform better than the rescoring results. Among these results, the best are the forward and SelectOne decoding results of CW & LA, with a CER of 5.59%. With the most optimal performing rescoring model as a criterion, the above models can achieve a 7.30% reduction in errors, proving their effectiveness.

Compared with the sentence ensemble model, the cumulative weight is significantly better than the majority weight. It is speculated that the method used in this paper can select words in the original model that are more in line with the textual meaning of the language model, thus correcting more errors and reducing the CER. Also, the alignment results are not much different. It is speculated that the decoding results of the five models show that there are fewer missing words in the sentence, and the positions of the words are roughly correct, so the effect of alignment is small. Among different models, some perform better in forward search and some perform better in backward search. So the SelectOne method can be used as the final result. SelectOne produces the most accurate results for this model. The guess is that SelectOne can take into account the semantics of the entire sentence. It also matches the possible number of characters in the entire sentence, so it can get closer to the correct result.

Table 2. CER of Sentence Ensemble

Model	Forward	Backward	SelectOne
MW & SA	5.97 %	5.96 %	5.95 %
CW & SA	5.62 %	5.61 %	5.60 %
MW & LA	5.93 %	5.95 %	5.93 %
CW & LA	**5.59 %**	5.60%	**5.59%**

4. CONCLUSION AND FUTURE WORK

From the experimental results, it can be seen that sentence ensemble can achieve lower CER when compared with the rescoring result, with the best result of 7.30% error reduction. The weight accumulation method in the model is better than the majority weight method, partly because the former is more in line with the semantics of the language model. And the "SelectOne" method can pick a more suitable sentence from two candidates to improve the accuracy.

The difficulty encountered by the current sentence ensemble method is that it uses rescoring results. This method can only find better word representatives from words that have already been decoded. Therefore, it cannot generate words that have not yet been decoded. Furthermore, the proposed method uses only five models. By using more models with significant differences, the choice of words can be more diverse to improve the accuracy. In addition, word selection only considers the weight of the majority and the design of perplexity. And the "SelectOne" method only considers the number of characters and PPL in the sentence. Therefore, if we continue to improve the method of word selection or SelectOne in the future, we may also achieve better results in speech recognition.

5. ACKNOWLEDGEMENTS

This work was supported in part by the E.SUN Financial Holding CO., LTD. of Taiwan, and the National Science and Technology Council, Taiwan (Grant no.: NSTC 110-2634-F-002-050-).

6. REFERENCES

[1] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur, "Recurrent neural network based language model.," in *Interspeech*. Makuhari, 2010, vol. 2, pp. 1045–1048.

[2] Kazuki Irie, Albert Zeyer, Ralf Schlüter, and Hermann Ney, "Language modeling with deep transformers," *arXiv preprint arXiv:1905.04226*, 2019.

[3] Martin Sundermeyer, Zoltán Tüske, Ralf Schlüter, and Hermann Ney, "Lattice decoding and rescoring with long-span neural network language models," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[4] Shankar Kumar, Michael Nirschl, Daniel Holtmann-Rice, Hank Liao, Ananda Theertha Suresh, and Felix Yu, "Lattice rescoring strategies for long short term memory language models in speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 165–172.

[5] Sriparna Saha and Asif Ekbal, "Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition," *Data & Knowledge Engineering*, vol. 85, pp. 15–39, 2013.

[6] Lidia Mangu, Eric Brill, and Andreas Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[7] Spyridon Matsoukas, J-L Gauvain, Gilles Adda, Thomas Colthurst, Chia-Lin Kao, Owen Kimball, Lori Lamel, Fabrice Lefevre, Jeff Z Ma, John Makhoul, et al., "Advances in transcription of broadcast news and conversational telephone speech within the combined ears bbn/limsi system," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1541–1556, 2006.

[8] Gökhan Tür, Jerry H Wright, Allen L Gorin, Giuseppe Riccardi, and Dilek Hakkani-Tür, "Improving spoken language understanding using word confusion networks.," in *Interspeech*, 2002.

[9] Hainan Xu, Tongfei Chen, Dongji Gao, Yiming Wang, Ke Li, Nagendra Goel, Yishay Carmiel, Daniel Povey, and Sanjeev Khudanpur, "A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5929–5933.

[10] Ke Li, Daniel Povey, and Sanjeev Khudanpur, "A parallelizable lattice rescoring strategy with neural language models," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6518–6522.

[11] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA)*. IEEE, 2017, pp. 1–5.