Location-Based End-to-End Speech Recognition with Multiple Language Models

Zhijie Lin,[1] Kaiyang Lin,[2] Shiling Chen,[3] Linlin Li,[4] Zhou Zhao[1]
[1]College of Computer, Zhejiang University, [2]Department of Philosophy, Sun Yat-Sen University
[3]College of Economy, Zhejiang University, [4]alibaba group
linzhijie@zju.edu.cn, linky7@mail2.sysu.edu.cn, chenshiling@zju.edu.cn,
linyan.lll@alibaba-inc.com, zhaozhou@zju.edu.cn

Abstract

End-to-End deep learning approaches for Automatic Speech Recognition (ASR) has been a new trend. In those approaches, starting active in many areas, language model can be considered as an important and effective method for semantic error correction. Many existing systems use one language model. In this paper, however, multiple language models (LMs) are applied into decoding. One LM is used for selecting appropriate answers and others, considering both context and grammar, for further decision. Experiment on a general location-based dataset show the effectiveness of our method.

Introduction

The End-to-End approach, using Connectionist Temporal Classification (CTC), has gone into current state-of-the-art automatic speech recognition pipelines in recent years and achieved many remarkable results. Because terminology is always recognized to their homophones, especially when it comes to a certain area, LMs are also served as powerful utility to get reasonable answers during the process of decoding, which can greatly improve the accuracy of recognition. Generally, decoding applies prefix beam searching (Hannun et al. 2014b), a heuristic searching algorithm, with one N-gram or RNN-based language model(LM) to find the best-path sequence. However, because of complexity of cutting words of Chinese, Chinese LMs are usually character-based, not word-based. Although existing methods have achieved good performance, they are mainly based on characters, which may ignore the relationship between words.

In this paper, we specifically consider the problem of location-based ASR from the viewpoint of Deep Neural NetWork (DNN) model with CTC and language models. When asking for a taxi, we can describe our location by a call with no need to operate on our cell phones. To improve the accuracy of recognition of geographical terminology, we propose a decoding and scoring method using more than one language model, which utilizes both character-based LM, word-based LM and class-based LM to make better use of both context and grammar information and gain a higher accuracy. The class-based LM applies clustering to analyze the grammar structure of a sentence. The decoding process can be divided into two steps. The first step, based on characters, is to get rid of answers with obvious mispronunciation and grammatical mistakes by beam searching, and get a set of alternative results. The other step, based on words and classes, is to analyze both context and grammar structure, rate the alternative answers and select the most applicable result.

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Methodology

The system overview is shown in the Figure 1. Our method uses a CTC neural network model to learn the pronunciation of each Chinese character. The encoder of the network has several layers of deep Convolutional Neural Network (CNN), which are followed by stacked bidirectional Gated Recurrent Unit (GRU) layers with CTC (Graves, Mohamed, and Hinton 2013).

With the pretrained CTC model $M(x)$, we get a possibility sequence $P = \{p_1, p_2, p_3, ..., p_t\}$, where x is a single utterance, t is the number of frames in the utterance, p_i is a possibility distribution vector over the vocabulary.

In the first step, by applying prefix beam searching algorithm utilized by a character-based language model LM_1 that contains many geographical nouns, we can get an alternative set $S_1 = \{(a_1, s_{1,1}), (a_2, s_{1,2}), ..., (a_m, s_{1,m})\}$, where m is the beam size, a_i is an alternative answer, $s_{1,i}$ is the evaluated score in the step one. Let c_{ij} be the j-th character in a_i, Nc_i be the number of characters in a_i. Therefore, step one is tend to find an alternative set, but also get rid of obviously wrong answers. We choose the beam size carefully and hope that the correct recognition or nearly correct recognition will be selected and thrown into the alternative set S_1. During the beam searching, $s_{1,i}$ of a_i can be computed as follows (Hannun et al. 2014a):

$$s_{1,i} = log(p(a_i|x)) + \alpha * LM_1(a_i) + \beta * Nc_i \quad (1)$$

Where

$$LM_1(a_i) = log(p(c_{i,1}, .., c_{i,Nc_i})) \quad (2)$$

The reason why we pick up a character-based language model as LM_1 is demonstrated by the formulas. We don't have to look ahead when computing $LM_1(a_i)$ above words sequence, so we can compute $s_{1,i}$ while moving cursor at p_i. However, character-based LM can't make full use of context like word-based LM, so we try to get a set S_1 instead of picking the answer with optimal score up.

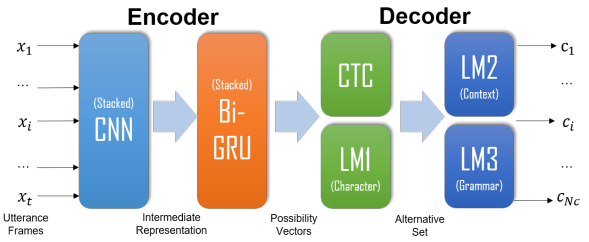

Figure 1: The system overview: a Hybrid CTC Network with Encoder, CTC Decoder and Multiple LMs. The possibility vectors are passed to Decoder with two steps.

In the second step, word-based and class-based LMs are applied into finding the optimal answer in S_1. The advantage over step one is that we own the whole sentence and context instead of decoding one by one. Let w_{ij} be the j-th word in a_i, Nw_i be the number of words in a_i, LM_2 be a word-based LM, $norm()$ be the value normalization, so the context score of a_i can be computed as follows:

$$context_i = \gamma * norm(s_{1,i}) + \delta * norm(LM_2(a_i)) + \epsilon * norm(Nw_i) \quad (3)$$

Where

$$LM_2(a_i) = log(p(w_{i,1}, .., w_{i,Nw_i})) \quad (4)$$

We add geographical nouns into the training of LM_2 once again to encourage the occurance of geographical nouns and reinforce the ability of geographical nouns recognition.

Then, inspired by some works of Natural Language Process (NLP), we first use K-Means clustering algorithm to classify the words into a specific group according to pre-trained word embeddings (Cha, Gwon, and Kung 2017). For example, verb and noun will be classified into two different groups. Therefore, we map a sentence $\{w_1, w_2, .., w_n\}$ into $\{group(w_1), group(w_2), ..., group(w_n)\}$, which greatly reduces the number of classes. Secondly, we can use Recurrent Neural Network (RNN) and the group training data to the modeling of grammar information. Let the grammar model be LM_3 that is class-based and the grammar score can be computed as follows:

$$grammar_i = \eta * norm(LM_3(group(w_{i,1}), .., group(w_{i,Nw_i}))) \quad (5)$$

Where

$$LM_3(a_i) = log(p(group(w_{i,1}), .., group(w_{i,Nw_i}))) \quad (6)$$

The final score s_i that gives a consideration of both context and grammar of a_i can be computed as follows:

$$s_i = context_i + grammar_i \quad (7)$$

Experiments and Results

Datasets We constructed a dataset that contains many location-based dialog utterances and was recorded by more than 200 people. The performance of our method was evaluated based on the test (20 hours) and development set (50 hours).

Language models	dev	test
No LM	18.16	17.05
LM_1	9.51	8.66
LM_1+LM_2	7.98	7.07
LM_1+LM_2+LM_3	7.69	6.78

Table 1: Character error rate (CER) with different language models on our dataset.

Implementation Details The LM_1, LM_2 employed N-gram based on corpus from Wikipedia, Baidu and our proper nouns library, and LM_3 employed RNN with 128 hidden cells. To combine different LMs, the parameter α was set to 2.6, β to 5.0, γ to 0.31, δ to 0.36, ϵ to 0.27, η to 0.09. In ASR, there are always some sentences that are correct grammatically but incorrect literally, so to make a balance, we didn't allocate a higher value to η.

Results As shown in the Table 1,D The CERs for no language model (No LM), character-based LM (LM_1), multiple LMs considering only context (LM_1+LM_2), and multiple LMs considering both context and grammar (LM_1+LM_2+LM_3) were shown, where LM_1+LM_2+LM_3 got best performance, showing that combining multiple LMs can effectively picks the more reasonable answers up from an alternative set.

Conclusion

In this paper, we present a new way of combining multiple LMs based on characters, words and classes to enhance the recognition accuracy of geographical nouns. This way can capture not only the context information through relationship among words, but also grammar information through learning the structure of sentences. Better results of recognition can be achieved with the same CTC model, which demonstrates the effectiveness of our method.

References

Cha, M.; Gwon, Y.; and Kung, H. 2017. Language modeling by clustering with word embeddings for text readability assessment. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2003–2006. ACM.

Graves, A.; Mohamed, A.-r.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, 6645–6649. IEEE.

Hannun, A.; Case, C.; Casper, J.; Catanzaro, B.; Diamos, G.; Elsen, E.; Prenger, R.; Satheesh, S.; Sengupta, S.; Coates, A.; et al. 2014a. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Hannun, A. Y.; Maas, A. L.; Jurafsky, D.; and Ng, A. Y. 2014b. First-pass large vocabulary continuous speech recognition using bi-directional recurrent dnns. *arXiv preprint arXiv:1408.2873*.