An Efficient Approach of Language Model Applying in ASR Systems[1]

Ying Liu[1] and Xiaoyan Zhu[2]

State Key Laboratory of Intelligent Technology and System,
Department of Computer Science, Tsinghua University
Beijing China, 100084
[1] liu98@mails.tsinghua.edu.cn, [2] zxy-dcs@tsinghua.edu.cn

Abstract

Language model plays a pivotal role in large vocabulary speech recognition systems. Providing more syntactic and semantic information, high-level language models hold stronger ability in guiding the search process and hence optimizing the final result. But on the other hand, complex language models, compared with simple ones, usually introduce proportional computing workload that jeopardizes the system efficiency significantly. In this paper, a new approach of applying high-order language models to decoding process will be proposed, and experimental result in our Gallina system will be provided, which demonstrate that, with high-level language models, the system performance is steadily increased while no much more efficacy is being lost as expense.

Keyword: Language model, search, speech recognition.

1 Introduction

Language model, defining the structure of actual language, representing lexical, syntactic, and semantic information, plays a critical role in speech recognition, spelling correction, handwriting recognition, machine translation and etc [1], in which language models not only work on improving the recognition accuracy but also constraining the search space [2].

There are mainly two types of language models, Chomsky's formal grammar which restricts sentences in solid format, and statistical language models which illustrate the probabilistic relationships among a sequence of words, including n-gram models (e.g., skipping [3], clustering [4], variable n-gram [5], caching models [6]), latent semantic analysis (LSA) model [7], maximum entropy (ME) model [8].

Statistical language models can be simply categorized into simple models and complex models. While simple models, such as uni-gram and bi-gram, focus on local

[1] Supported by the National Natural Science Foundations of China No.60272019 and No.60321002.

information or less historical information, complex models include more global constraints or more history clues, exemplified by high-order n-gram models.

Though high-order models depict more linguistic information hence of benefit for performance, they also introduce more complexity and reduce the efficiency because of explosive search space. The case is that, with n-gram model, more historical words need to be considered, so the linguistic state can't be locally determined [1], and lexicon trees must be duplicated for various history states which will terribly expand the search space. For this reason, it's difficult to implement a high-order language model based recognition system, although some space-control techniques such as pruning and look-ahead algorithm [9] can be used to alleviate the problem to some extend.

Among current practical recognition systems, mainly two approaches have been utilized for language model applying: one is one-pass strategy [3],[9],[12],[13],[14], which combining the probability of acoustic and language models directly by simple multiplication, is less efficient for high-order language models, and another, multi-pass strategy [10],[15],[16], constructs a n-best list or a word graph in the first decoding pass using acoustic and simple language models, and then applies more powerful but expensive high-order models to pick out the best candidate. Considering the multi-pass strategy is not suitable for real-time applications, we are apt to select one-pass search but improve the efficiency in the case of high-order language models.

In this paper, we introduce a new approach named "partial-path-adjusting" for high-order language model, which demonstrates a trade off between precision and complexity. Getting the idea that high-order models can be used as partial candidate selector, one-pass algorithm in our Gallina system is modified to optimize the partial paths with historical information provided by the high-order linguistic score. The experimental result in Gallina system testified the validity of our approach, which, at the time of improving the recognition performance, did not introduce any apparent efficiency reduction.

The organization of this paper is as follows. In sections 2, we first review both the one-pass and multi-pass decoding strategy, and then our approach will be discussed in detail in section 3, which is followed by our experiment results in section 4, and at last, in section 5 we get some conclusions.

2 Specification of the Search Strategy

2.1 The Role of Language Models in Speech Recognition

In most practical ASR systems, the destination of continuous speech recognition is to find the best matching of the observation sequence x_1^T and a word sequence w_1^N based on maximizing the posterior probability. Applying Bayes' decision rule and Viterbi approximation [6], the best word sequence can be denoted as the following form:

$$[w_1^N]_{opt} = \arg\max_{w_1^N}\{P(w_1^N)\cdot P(x_1^T \mid w_1^N)\} \cong \arg\max_{w_1^N}\left\{P(w_1^N)\cdot \max_{s_1^T} P(x_1^T, s_1^T \mid w_1^N)\right\} \quad (1)$$

where $P(x_1^T \mid w_1^N)$ is an acoustic probability, which denotes the possibility that a series of words issue the observation sequence, and $P(w_1^N)$, probability given by a language model, describes the occurrence possibility of this word series itself.

2.2 Time-synchronous Search Strategy

Time-synchronous Viterbi beam search is an efficient search algorithm used in most speech recognition systems. In this kind of decoding process, complex language models can be handled in two manners, one-pass and multi-pass strategy.

2.2.1 Time-synchronous one-pass Viterbi-beam Algorithm

Suppose the observation sequence and the optimized partial word sequence is:

$$\underbrace{x_1 x_2, \ldots, x_\tau}_{w_1 \ldots w_{n-1}} \underbrace{x_{\tau+1} \ldots x_t x_{t+1} \ldots x_\zeta}_{w_n} \underbrace{x_{\zeta+1} \ldots x_T}_{\ldots\ldots} \tag{2}$$

Assume the predecessor is v ($w_{n-1} = v$) and the word boundary of v is frame τ , the score of path L, which stagnates in state s_t at frame t, is[2,9]:

$$H(t, s_\mathrm{t}; \tau, w_1^{n-1}) = G(w_1^{n-1}; \tau) \cdot Q_v(t, s_\mathrm{t}; \tau)\,, \tag{3}$$

The first part of (3) can be defined as

$$G(w_1^{n-1}; \tau) = P(w_1^{n-1}) \cdot \max_{s_1^\tau} P(x_1^\tau, s_1^\tau \mid w_1^{n-1}), \tag{4}$$

which denotes the combination score of partial path L at frame τ ,derived from acoustic and language models, and can be named as the history score of L .

The rear part of (3) can be rewritten as

$$Q_v(t, s_\mathrm{t}; \tau) = \max_{s_{t-1}} \{Q_v(t-1, s_{t-1}; \tau) \cdot P(x_t, s_t \mid s_{t-1})\}, \tag{5}$$

which represents the acoustic probability of the optimal state sequence $s_{\tau+1}^t$ from frame $\tau+1$ to frame t , and it can be regarded as the acoustic score of path L at frame t.

The probability of language model will be applied at frame ζ because the path L reaches the word boundary of w_n at that time, as shown in (2). Although several different paths reach that boundary all together, only the best partial hypothesis with the highest score will be reserved [2],[9].

$$H(w_1^n; \zeta) = \max_{w_1^n} \{(G(w_1^n .; \zeta)\} = \max_{w_1^{n-1}} \{P(w_n \mid w_1^{n-1}) \cdot \max_\tau (G(w_1^{n-1}; \tau) \cdot Q_v(\zeta, s_\zeta; \tau))\}. \tag{6}$$

With proper and efficient pruning techniques, time-synchronous one-pass Viterbi-beam strategy is fit for real-time applications, but when high-level language models are considered, more deliberately designed evaluation functions are needed to balance the precision and efficiency.

2.2.2 Time-synchronous Multi-pass Search Strategy

The concept of multi-pass search is introduced for applying more linguistic information within a reasonable search space. In the early passes, more discriminate and computationally affordable models are used to reduce the number of hypotheses and get a couple of reasonable candidates to construct a word graph, and then, more powerful and expensive models are applied in the subsequent passes, to search in the word graph until the optimal word sequence is reached.

There are no constraints of the algorithms and knowledge sources in multi-pass search. The popular way is using bi-gram model, context-independent acoustic models and time synchronous Viterbi-beam search in the first pass, while tri-gram and other more complicated language models, context-dependent acoustic models and time asynchronous stack decoding method are used in the later passes to find out the optimal result.

Multi-pass strategy can handle many complicated acoustic models and language models conveniently. Although there are a few inadmissible pruning errors with each earlier passes, the experiments proved that the result of multi-pass search is better than one-pass strategy. The most important criticism of multi-pass search is that they are not suitable for real-time applications, because no matter how fast the first pass is, the successive passes cannot start until users finish speaking.

3 An efficient Approach of High-order Language Model Applying

We examined the idea of re-ordering the paths using complicated language models in multi-pass search, and try to incorporate the probabilities of complex language models dynamically into one-pass search. Without expanding the search space and reducing the efficiency, in our “partial-path-adjusting” algorithm, the simple language model is used to direct the search, and at this time, complicated models are used to optimize the score of the partial hypotheses so that more reasonable results can be achieved.

The proposed algorithm can be described as the following:

1. Compute each possible partial path’s score H in the search space. Using Viterbi-beam one-pass algorithm with simple language models, the joint probability of partial path L is:

1) If L doesn’t reach a word boundary at frame t,

$$H(t, s_t; \tau, w_1^{n-1}) = G(w_1^{n-1}; \tau) \cdot Q_v(t, s_t; \tau) = P(w_1^{n-1}) \cdot \max_{s_1^\tau} P(x_1^\tau, s_1^\tau \mid w_1^{n-1}) \cdot Q_v(t, s_t; \tau) \qquad (7)$$

2) If L reaches a word boundary at frame t,

$$H(w_1^n; t) = \max_{w_1^n}\{(G(w_1^n; t)\} = \max_{w_1^{n-1}}\{P(w_n \mid w_1^{n-1}) \cdot \max_{\tau}(G(w_1^{n-1}; \tau) \cdot Q_v(t, s_t; \tau))\} \qquad (8)$$

where $P(w_n \mid w_1^{n-1}), P(w_1^{n-1})$ are derived from simple language models; Actually, (7) and (8) are rewritten from equation (3) and (6) with different parameters , the details can be found from Section 2.2.1.

2. Sort all paths according to their joint probabilities H and select the top N paths denoted as a set Ψ in order to reduce unnecessary computation in the later steps. In fact, since the paths that hold higher scores have more influences on the final result, it is not necessarily to adjust each possible path’s score.

3. Use high-level language model to compute $P_{high-order}(W)$ as the adjusting weight

for each path in Ψ. The expression of $P_{high-order}(W)$ should correspond with the definition of the high-level language model itself.

4. Adjust the paths in Ψ. The formula (7) and (8) are modified in this step to re-compute the partial paths' scores according to $P_{high-order}(W)$.

1) If L doesn't reach a word boundary at frame t, the re-scored probability is,

$$\begin{aligned} H(t,s_{\mathrm{t}};\tau,w_1^{n-1}) &= G(w_1^{n-1};\tau)\cdot Q_v(t,s_{\mathrm{t}};\tau)\cdot P_{high-order}(w_1^{n-1}) \\ &= P(w_1^{n-1})\cdot \max_{s_1^\tau} P(x_1^\tau, s_1^\tau \mid w_1^{n-1})\cdot Q_v(t,s_{\mathrm{t}};\tau)\cdot P_{high-order}(w_1^{n-1}), \end{aligned} \tag{9}$$

2) If L reaches a word boundary at frame t, the re-scored probability is,

$$H(w_1^n;\mathrm{t}) = \max_{w_1^{n-1}}\{P(w_n \mid w_1^{n-1})\cdot \max_{\tau}(G(w_1^{n-1};\tau)\cdot Q_v(\zeta,s_{\mathrm{t}};\tau)\cdot P_{high-order}(w_1^n))\}, \tag{10}$$

5. Extend all paths (both the paths in Ψ and outside Ψ) to the next frame and repeat the process.

In our experiment, bi-gram model is used as the simple language model to compute the word-level score, and tri-gram probabilities are used to adjust those scores. In the case of tri-gram, $P_{high-order}(W)$ is defined as:

$$P_{trigram}(w_n) = \frac{p(w_1)\cdot p(w_2 \mid w_1)\cdot ....\cdot p(w_{n-1} \mid w_{n-3}w_{n-2})}{n-1}, \tag{11}$$

The denominator (n-1) in (11) is introduced to eliminate the effect of sentence length. To overcome the inherent data sparse problem of tri-gram model, a linear interpolation smoothing is used as the following:

$$p(w_i \mid w_{i-2}w_{i-1}) = \lambda_1 p(w_i \mid w_{i-2}w_{i-1}) + \lambda_2 p(w_i \mid w_{i-1}) + \lambda_3 p(w_i) + \lambda_4 / N \tag{12}$$

where N is the total word number in lexicon.

If the high-level language model is defined as Latent Semantic Analysis model or other more complicated models, the adjusting weight $P_{high-order}(W)$ should be modified according to the concept of language model.

In the following paragraphs, as an example, the processing result of a Chinese word sequence is discussed based on bi-gram and tri-gram models.

Assume one observed sequence, whose standard transcription is: 交 所在 地 主管 部门 审批 (In English, this sentence means: deliver something to the local departments who are in charge of it.), is fed into our syllable based decoder, and two partial paths L_1, L_2 reach frame t as shown in Fig.1. L_1 represents the actual partial path of the observed sentence, while L_2 has no real-life meaning in Chinese as a whole, but each word of L_2 has some particular relationships with its neighbors. These two paths have various history information of w_1^{n-1}, and get to the same word boundaries of various predecessors at frame τ. At frame t, the path pair L_1, L_2 stay in state s_t and none of them reached any word boundary from frame $\tau+1$ to frame t (include t).

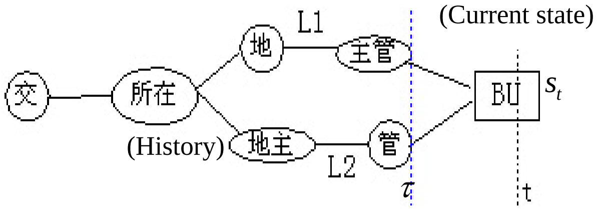

Fig. 1. Two partial paths in the search space when dealing with the observed sentence: 交 所在 地 主管 部门审批.

Applying the “partial-path-adjusting” algorithm to L_1 and L_2, the results are listed step by step as following.

1. Compute the original paths’ scores of L_1, L_2 at frame t with bi-gram model and the equation (7).

$$H_L(t,s_t;\tau,w_1^{n-1}) = G(w_1^{n-1};\tau)\cdot Q_v(t,s_t;\tau) = P(w_1^{n-1})\cdot \max_{s_1^\tau} P(x_1^\tau, s_1^\tau \mid w_1^{n-1})\cdot Q_v(t,s_t;\tau), \quad (14)$$

2. Sort the scores of the two paths. In this instance, L_1, L_2 have the same parameters of t, s_t and τ, but the history word sequences w_1^{n-1} are different, therefore, the bi-gram language model probabilities are very important. Due to the inherent limit of bi-gram model itself, the bi-gram probabilities based on our lexicon and training corpus is

$$p(\text{地}\,|\,\text{所在}) + p(\text{主管}\,|\,\text{地}) < p(\text{地主}\,|\,\text{所在}) + p(\text{管}\,|\,\text{地主}) \quad (15)$$

so the result of step 2 is: $H_{L_2}(w_1^n, t) > H_{L_1}(w_1^n, t)$, (16)

Without the “partial-path-adjusting” algorithm, when these two paths extend forward and reach the boundary of the same word “部门”, the optimization, according to the equation (8), will cast away L_1 because its joint probability is lower than L_2.

3. Compute $P_{trigram}(w_n)$ using the equations (10) and (11) to check more history information and the relationship of the two adjusting weights are

$$P_{trigram}(\text{所在\_地\_主管}) > P_{trigram}(\text{所在\_地主\_管}) \quad (17)$$

Obviously L_1 is more reasonable than L_2 in the meaning of natural language.

4. Adjust the original paths’ scores of L_1, L_2 with $P_{trigram}(w_n)$, and $H_{L_2}(w_1^n, t)$ is penalized so that more reasonable candidate, L_1, is reserved.

4 Experiment Result

Our experimental platform, Gallina, is a syllable-based speaker-independent large vocabulary continuous speech recognition system. Similarly to the popular practice ASR systems, Gallina uses 39-dim MFCC as frame features, and multi-mixture density CHMM as acoustic models. The time-synchronous one-pass Viterbi-beam search strategy and token passing algorithm are used for decoding. Techniques including phoneme look-ahead, language model look-ahead are used to speed up the search procedure, and certain pruning tips including acoustic pruning, language model pruning, histogram pruning and tree pruning are used to delete some unpromising hypotheses.

The acoustic models are trained on 863 Chinese continuous speech database , and the bi-gram model is built on the text corpus in size of two million characters, compared with the tri-gram model that are built on the text data of two hundred million characters. The size of lexicon is up to 6,3995 that is built upon the one-level and two-level Chinese word databases. At last, Katz back-off smoothing is used for bi-gram model and linear interpolation smoothing is used for tri-gram.

All experiments were performed on a training set of 6264 utterances from 12 speakers and evaluated by 740 utterances from 2 speakers, with all recognition parameters optimized.

In the first set of experiments, we focused on the recognition results of the “partial part adjusting” algorithm with several different language models. The experiment results are shown in Table 1.

In table 1, the “baseline” system was built without any language model information, and the results which represented in the format of “syllable string” were specified by the “Acoustic correct rate”. The “bi-gram” system was referring to bi-gram model only, while the “bi&tri-gram” system was using our “partial-path-adjusting” algorithm and combining bi-gram and tri-gram model together. The “word string” results of these three systems mentioned above were specified by “Word Correct, Substitution, Deletion, Insertion, Accuracy, and Error rate”. From these results, it can be understood that using complicated language models in the way of “partial-path-adjusting” contributes fairly well to the performance of speech recognition system, in our case, by 3% reduction in “Word Error Rate” (WER).

Table 1. Recognition results of the “partial-path-adjusting” algorithm. (Correct= Accuracy+ Insertion, Error= Substitution+ Deletion+ Insertion)

	Baseline	Bi-gram	Bi-&Tri-gram
Acoustic Correct	70.6%	78.47%	79.37%
Word Correct	-----	62.24%	64.96%
Word Substitution	-----	36.63%	33.99%
Word Deletion	-----	1.13%	1.04%
Word Insertion	-----	9.67%	9.37%
Word Accuracy	-----	52.57%	55.6%
Word Error	-----	47.43%	44.4%

Table 2. Average performance of the “partial-path-adjusting” algorithm with different language models on single CPU.

	Bi-gram	Bi&Tri-gram
Average Time (sec)	17.32	15.99

Another experiment is set up to check the performance of our algorithm on the CPU of an Athlon64 3200+. The results are shown in Table 2 where “Average Time” presents the average recognition time consumption of each sentence throughout the whole recognition processing. We find out that using high-order language model directly in the first pass of speech recognition improved system efficiency slightly, since the partial-path-adjusting technique enhanced the accuracy of the predecessors, some unlikely partial paths would be destroyed earlier and the total computation would be reduced.

5 Conclusion

In this paper, we present a new approach of applying high-order language model into speech recognition tasks. The experiment results proved that this approach enhanced the system performance without apparent hazard on system efficiency. And although tri-gram is the case in our experiment, this algorithm is not customized specially to N-gram models, so other complicated language models such as LSA can also be applied in the same way.

References

[1] Joshua T.Goodman: A bit of progress in language modeling. Computer speech and language, Vol. 15 (2001), pp. 403-434.

[2] Hemann ney, Stefan ortmanns: Dynamic programming search for continuous speech recognition. IEEE Signal Processing Magazine (1999), pp. 64-83.

[3] Xuedong Huang, Fileno Alleva, Hsiao-Wuen Hon, Mei-Yuh Hwang, Kai-Fu Lee and Ronald Rosenfeld: The SPHINX-II Speech Recognition System: an Overview. Computer Speech and Language, Vol. 2 (1993), pp. 137-148.

[4] Martin S., J. Liermann, et al: Algorithms for Bigram and Trigram Word Clustering, Speech Communication, Vol. 24 (1) (1998), pp. 19-37.

[5] Siu, M.,M. Ostendorf: Variable N-grams and Extensions for Conversational Speech Language Modeling. IEEE Trans. on Speech and Audio Processing, Vol. 8(1) (2000), pp. 63-75.

[6] Kuhn,R., R.D.Mori: A Cache-Based Natural Language Model for Speech Recognition. IEEE Trans.on Pattern Analysis and Machine Intelligence, Vol. 12 (6) (1990), pp. 570-583.

[7] Bellegarda,J.R.: Exploiting Latent Semantic Information in Statistical Language Modeling. Proceedings of the IEEE, Vol. 88 (8) (2000), pp. 1279-1296.

[8] Berger, A. L., S. A. D. Pietra, et al: A Maximum Entropy Approach to Natural Language Processing, Computational Linguistics, Vol. 22 (1) (1996), pp. 39-71.

[9] Stefan ortmanns ,Hermann ney: Look-ahead techniques for Fast Beam Search. Computer speech and language, Vol. 14. (2000), pp. 15-32.

[10] Richard Schwartz , Yen-lu chow: The N-best algorithms: an efficient and exact procedure for finding the N most likely sentence hypotheses. ICASSP (1990).

[11] F.Jelinek: Continuous speech recognition by statistical methods. Proceedings of the IEEE, Vol. 64 (10) (1976), pp. 532-556.

[12] H.Ney,L.Welling, S.Ortmanns, K.Beulen, F.Wessel: The RWTH Large Vocabulary Continuous Speech Recognition System. ICASSP (1998), pp. 853-856.

[13] Gauvain, J.L., Lamel,L.,Adda-Decker,M.: Developments in continuous speech dictation using the ARPA WSJ task. ICASSP (1995), pp. 65-68.

[14] Woodland,P.C., et al : Large Vocabulary Continuous Speech Recognition Using HTK. ICASSP (1994), pp. 125-128.

[15] H. Shu, C. Wooters, O. Kimball,T. Colthurst, F. Richardson, S. Matsoukas and H. Gish: The BBN Byblos 2000 conversational Mandarin LVCSR system , Proceeding 2000 Speech Transcription Workshop.
[16] Andrej Ljolje, Michael D. Riley: The AT&T Large Vocabulary Conversational Speech Recognition System. Eurospeech (1999).

Ying Liu, Master Student of Department of Computer Science and Technology, Tsinghua University. BSc, Univerisity of Computer Science and Technology, Tsinghua University, 1998.

Xiaoyan Zhu, Professor, Deputy Head of Department of Computer Science and Technology, Tsinghua University. BSc, University of Science and Technology Beijing, 1982; MSc, Kobe University, Japan, 1987, Ph. D., Nagoya Institute of Technology, Japan, 1990.