EuroSpeech, 3:2139-2142, Sept. 3-7, 2001, Aalborg, Denmark

An Online Incremental Language Model Adaptation Method

Genqing Wu, Fang Zheng, Ling Jin, and Wenhu Wu

Center of Speech Technology, State Key Laboratory of Intelligent Technology and Systems
Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China
[wgq, fzheng, jinl]@sp.cs.tsinghua.edu.cn, http://sp.cs.tsinghua.edu.cn

Abstract

In this paper, an online incremental language model adaptation method is proposed, which is different from the traditional offline language model adaptation method. There are some problems in the online incremental adaptation. The first one is how to adjust the model parameters online and modify the model incrementally. The second one is how to induce new words and assign initial probabilities to the n-grams related to them. In our application for Chinese character input method editor, the language model is divided into two parts, corresponding to the background (general-purpose) model and the user model, respectively. A modified maximum a posterior method is proposed for adapting the user model dynamically. Experiments are done to test the proposed method on a Chinese sentence input system and the results show that a satisfying word error rate reduction is obtained when the input articles are of similar topics.

1. Introduction

The statistical language model (LM) is widely used in automatic speech recognition applications. Usually the language model is trained from a large text corpus and then some re-estimation methods, such as *discounting*, *backing-off* and *interpolation*, are used to deal with the data sparseness problem [1]. The collection of general domain training sentences/texts is time-consuming, expensive and difficult, and the trained LM is impossibly suitable for every user, therefore language model adaptation (LMA) techniques are quite necessary.

The LMA is usually conducted by combining a general-purpose well-trained model with a domain-oriented poor-trained model, which is often called the *topic adaptation* or *domain adaptation*. For example, there are infinite documents of all domains in the world, many of which can be used to train a general-purpose model *M.* Given the sample data *S* in the target domain, the task of LMA is to produce a model for the particular domain by using *S* and useful information from *M*. Using such methods can improve the LM performance more or less [2,3,4]. Nevertheless, almost all of them are in an off-line non-incremental manner so it is difficult and inconvenient to use them in real-time applications such as the Chinese input method editor (IME).

Several techniques can be used for the LM adaptation, such as maximum a posteriori (MAP) [4], maximum entropy, minimum discrimination information (MDI) [3,5,6], and so on.

The MAP method takes the general-purpose model for the prior probability estimation, and tries to combine two or more models using the linear interpolation, which makes the adaptation a selection of appropriate interpolation parameters. The Bayesian and expectation-maximization algorithms are proposed in this situation and achieve good performances [7].

The primary emphasis of MDI method is to estimate the new language model *as close as possible* to the general-purpose language model and meet the constraints derived from the relatively small adaptation data in the particular domain. Some methods are proposed to measure this *close degree* between the background language model and the adapted language model. The *Kullback-Leibler* distance measure is one of the most frequently used techniques. At first it has been found that the interpolation is an efficient method, and gradually there has been evidences showing that exponential models are superior to the linear interpolation [8]. But all of them need to use general iterative scaling, which costs a lot of computation.

No matter how better or worse all the above methods work, almost all of them are designed to orient a particular domain and typically the adaptation corpus should be collected in advance. In case that the adaptation corpus is relatively too small to build a perfect LM, such work cannot be done offline before the LM is being used in particular applications. For example, in the Chinese character IME, the adaptation corpus is inconvenient to collect in advance. As a matter of fact, the adaptation corpus can only be collected sentence by sentence when the user inputs Chinese sentences. Obviously, a straightforward idea is that the LM is adapted online and incrementally, in other words, sentence by sentence.

To perform the online incremental language model adaptation, an appropriate language model structure is designed to enable the online modification of some language model parameters. Furthermore, the whole model is divided into two parts, one is the fixed general purpose model, called the *background model*, which can be trained offline using a large corpus, the other is the user-oriented model, called the *user model*, which is generated and adapted online according to the user's sentence-by-sentence input. Strategies for the parameter adjusting and a new word induction method will also be presented based on such a structure. Finally experiment results will be given.

2. Language Model Adaptation

As mentioned above, the whole model consists of two sub-models. Actually, the general-purpose model is a perfect one and it can be used individually, while the user model just captures adapted language phenomenon when being used and it is only a supplement of the general-purpose one.

2.1. Design of General-Purpose Language Model

In the trigram language modeling, the probability of given a sentence, i.e. word string, $S = w_1 w_2 ... w_n$ is calculated as

$$P(S) = P(w_1)P(w_2|w_1)\prod_{i=2}^{n} P(w_i \mid w_{i-1}, w_{i-2}) \tag{1}$$

The goal of the Chinese sentence IME is to find the best Chinese sentence $\hat{S}$ given the input pinyin string X

$$\hat{S} = \arg\max_{S} P(X \mid S)P(S) \tag{2}$$

The trigram language model suffers from the serious data sparseness problem and several methods are proposed for LM smoothing, such as *discounting*, *backing-off* and *interpolation*. In our language framework, the Katz smoothing method is adopted [1] as follows

$$P_{Katz}(w_i \mid w_{i-2}, w_{i-1}) = \begin{cases} C(w_{i-2}, w_{i-1}, w_i)/C(w_{i-2}, w_{i-1}), \\ \qquad\qquad if \;\; r > r_T \\ d_r C(w_{i-2}, w_{i-1}, w_i)/C(w_{i-2}, w_{i-1}), \\ \qquad\qquad if \;\; 0 < r \le r_T \\ \alpha(w_{i-2}, w_{i-1}) P_{Katz}(w_{i-1} \mid w_i), \\ \qquad\qquad if \;\; r = 0 \end{cases} \tag{3}$$

where $C(\cdot)$ means the occurring count of the specified event, r is an occurring count value, r_T is a count threshold for discounting purpose, $\alpha(w_{i-2}, w_{i-1})$ and d_r are the smoothing parameters, see [1] for more details. It is assumed that $\alpha(w_{i-2}, w_{i-1})$ and d_r are unchanged during the adaptation, this assumption is reasonable because the adapted data is smaller -- by several orders of magnitude -- than the training data.

In order to accelerate the access to the background model, we propose some techniques to speed up the decoding procedure dramatically [9]. For the sake of adaptation, we store the occurring counts, instead of the estimated probabilities, of trigrams in our background model; this structure is more helpful for the online adaptation.

2.2. MAP Estimation and Adaptation

If denoting $P(w \mid h) = \lambda_{hw}$, the language model parameter set can be written as

$$\phi = \left\{ \lambda_{hw} \mid w \in W, h \in H \right\} \tag{4}$$

where W is the set of all possible words $\{w\}$ and H is the set of all possible histories $\{h\}$. The adaptation is to re-estimate the parameters in ϕ after the sample data X is observed. The goal of the MAP estimation is to find the parameter estimation ϕ_{MAP} that maximizes the posterior probability $P(\phi|X)$, so we have

$$\phi_{MAP} = \arg\max_{\phi} P(X \mid \phi)P(\phi) \tag{5}$$

where X is the sample data used for adaptation and $P(\phi)$ is the prior probability distribution of parameter ϕ, the widely used *Derichlet* distribution is a good assumption for it

$$P(\phi) = k \prod_{w \in W} \prod_{h \in H} \lambda_{hw}^{\alpha_{hw} - 1} \tag{6}$$

where k is a constant used for the normalization purpose, and α_{hw}'s are hyper-parameters usually estimated from the gram (hw) occurring count $C_{hw}^{(T)}$ in the training data as

$$\alpha_{hw} = C_{hw}^{(T)} + 1 \;\; \left(w \in W, h \in H \right) \tag{7}$$

As shown in Equation (3), $\lambda_{hw} \overset{\Delta}{=} P(w|h)$ is calculated using the maximum likelihood (ML) method

$$\lambda_{hw} = \frac{C_{hw}}{\sum_{w \in W} C_{hw}} \tag{8}$$

Given the adaptation text X, if $C_{hw}^{(A)}$ is the occurring count of the gram (hw), we have

$$P(X \mid \phi) = \prod_{w \in W} \prod_{h \in H} \lambda_{hw}^{C_{hw}^{(A)}} \tag{9}$$

According to Equations (6), (7), and (9), Equation (5) can be rewritten as

$$\phi_{MAP} = \arg\max_{\phi} \; k \prod_{w \in W} \prod_{h \in H} \lambda_{hw}^{C_{hw}^{(T)} + C_{hw}^{(A)}} \tag{5'}$$

Considering the normalization constraint

$$\sum_{w \in W} \lambda_{hw} = 1 \tag{10}$$

the MAP estimation result can be written as

$$\lambda_{hw}^{MAP} = \frac{C_{hw}^{(T)} + C_{hw}^{(A)}}{\sum_{w \in W} \left(C_{hw}^{(T)} + C_{hw}^{(A)} \right)} \tag{11}$$

Refer [10] for more information. Using the MAP method, the difficulty shifts to finding a way to adjust the occurring counts of n-grams according to the single input sentence in each adaptation procedure. When we receive a pinyin string X from the user, we first convert it into a Chinese word string (also can be regarded as a character string) S and then output it, which may be modified into a *correct* string S_M by the user. Both X and S_M are used for the incremental adaptation. Here three primary principles are adopted.

- The adaptation should correct the conversion errors introduced by the background language model as many as possible.
- In order to avoid the unnecessary and fatal damage to the background LM, the adaptation (extent and speed) should be moderate enough.
- The adaptation should be quick enough to work online

Accordingly, we use the following approaches.

- Suppose the conversion result in the word string form is $w_1...w_{n-1}w_n$ and the user modifies some words, but the percentage of modified words is small due to the high-accuracy of the language model. Because most words have been correctly converted, only the counts of those n-grams that appear in the sentence and contain modified words need to be updated.
- The new corpus should be emphasized by a weight α because the adaptation corpus is too small, therefore Equation (11) should be modified as

$$P(w \mid h) = \frac{C_{hw}^{(T)} + \alpha * C_{hw}^{(A)}}{\sum_{w \in W} \left(C_{hw}^{(T)} + \alpha * C_{hw}^{(A)} \right)} \qquad (12)$$

The parameter α changes dynamically with different input sentences. Obviously, with the increase of α, the correct sentence will have a bigger probability to be the first candidate. Suppose α_{good} is the threshold according to which the correct sentence will be converted, α can be chosen with the following formula

$$\alpha = \begin{cases} \alpha_{\min}, & if \ \ \alpha_{\text{good}} \le \alpha_{\min} \\ \alpha_{\text{good}}, & if \ \ \alpha_{\min} < \alpha_{\text{good}} < \alpha_{\max} \\ \alpha_{\max}, & if \ \ \alpha_{\text{good}} \ge \alpha_{\max} \end{cases} \qquad (13)$$

where $\alpha_{\min}$ and $\alpha_{\max}$ are reasonable minimal and maximal values of α, respectively, they are determined empirically offline.

- Direct modification of the background model is time consuming, especially when we add new words and hence n-grams. In order to avoid the memory data movement in the background model, if the adapted n-grams are found in the background model, their counts will be modified and stored in the user model only. Hence such n-grams can be seen in both models with different count values, where counts in the user model take precedence. The user model also stores new grams that cannot be found in the background one. Its smaller size makes the access to it quicker. On the other hand, the separate storing of the two models guarantees that the whole model do not diverge too far away from the background one and makes it possible to forget less and less frequently seen *grams*. That the n-grams with error words in the converted sentence are found in the user model indicates that these errors may be caused by the user's previous input. In this case the occurring counts of those grams should be removed from the user model. In our solution, these counts will be halved; grams with smaller count values than those in the background model will also be discarded.

2.3. New Word Induction

Usually, the above method works well if there are not too many errors in one converted sentence, but the performance degrades if many errors are found, especially when incorrect-converted words are adjacent. This is often caused by new word(s). A method should be used to induce new word(s) online, which is different from the complex off-line word induction.

Because each Chinese word is consisting of one or several Chinese characters, a Chinese sentence can also be regarded as a character string. Suppose there are L (L>1) successive improper characters in the conversion result and the maximum length of a proper word is L_{max}, we have the following different cases.

- If $L < L_{max}$ and the improper character sequence can be a whole word, just perform a normal adaptation.
- If $L \le L_{max}$ and all the improper characters can not form a multi-character word, add a new word consisting of these characters, and then perform a normal adaptation on this new word. (According to the previous subsection, incorrectly induced new words can be removed later automatically.)
- If $L > L_{max}$ and all the improper characters cannot form a multi-character word, segment the string into several parts whose lengths are all less than L_{max} such that the average α reaches its smallest value. Actually, this situation seldom occurs if the language model is of high accuracy.
- If $L > L_{max}$ but multi-character words can be found in this character string, segment this string into a sequence of words, each of which can be either a multi-character word or a single-character word. Then perform the above steps recursively.

We do the word induction using the above simple method and it works well, we also find that most of the induced words are person/place names or proper nouns.

2.4. Practical Considerations

Obviously, the adaptation on the whole language modeling is time-consuming. In order to speed up this procedure, we design our language model exquisitely. As stated above, we use two models, the background model is for general purpose while the user model for user purpose, and the structures of the two models are exquisitely designed and similar to each other (the counts instead of probabilities are stored). There are two kinds of grams in the user model. Some can be found in the background model, they are adapted ones. The others cannot be found in the background model, they correspond to the inducted new words and new grams. A flow chart is illustrated in Figure 1.

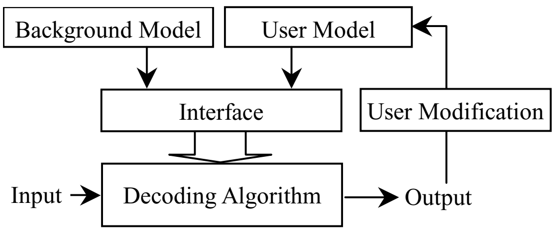

Figure 1. Model structure

The data structure for the background model is designed in an index style [9] as illustrated in Figure 2.

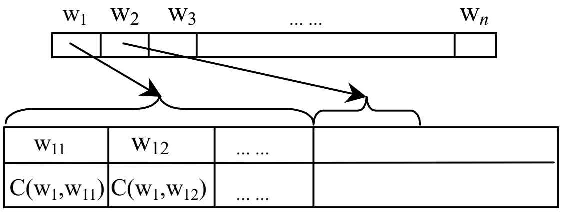

Figure 2. Bigram data structure.

The trigrams and smoothing parameters are organized similar to the bigrams. The count values in the user model may be very large after the system has been used for a long time, so it's necessary to avoid overflow. One efficient method is to decrease those counts of grams found in the decoded first candidate but modified by the user. Counts of those grams that are in the first decoded candidate and confirmed by the user will be kept as is.

A lexicon tree [11] of the whole vocabulary (including induced new words) is built to help decoding pinyin string into Chinese words. Considering that the word induction will cause the lexicon tree to be reorganized costly, we divide it into 26 sub-trees according to the first letter of pinyins [5].

3. Experiments

We collect a corpus of different topics and do our experiments based on it; results are shown in Table 1.

Table 1: Online incremental adaptation performance of several topics. (Topic A is the news about the president election in USA; Topic B is the news about the military maneuver of China; and Topic C is some miscellaneous news without a fixed topic.)(WER stands for word error rate.)

Topic	# of words	WER before adapt.	WER after adapt.	WER reduction
A	4,424	7.64%	6.19%	19.0%
B	3,602	9.08%	6.41%	29.4%
C	3,103	9.64%	9.60%	0.4%

As shown in Table 1, Topic C is a mixed topic and hence the adaptation performance on it is indistinctive. Further more, if we select independent sentences as test corpus, we will find that the adaptation performance degenerates because the information adapted from previous sentences may mislead the decoding of the successive sentences.

We also do some experiments to reduce the WER when we apply our adaptation method on one corpus for more than one time, the results are shown in Table 2.

Table 2: Adaptation performance when applying our method on one corpus for twice

Test Corpus	# of Words	WER Reduction
863 Corpus	23,310	47.4%
Training Corpus	30,441	29.2%

The WER reduction is not 100% because we limit the dynamic weight α with an upper bound, which guarantees the adaptation to be moderate enough.

4. Conclusion

In this paper an online incremental language model adaptation method based on modified MAP estimation has been proposed, using a dynamic weight to combine the background language model and the user model. It adjusts the parameters of language model efficiently, and avoids sharp changing to make the model stable. We use the word induction method to induce new words, especially new person/place names from the user's input. Experiments show that it can achieve a considerable improvement when the input articles are on similar topics. Actually, our adaptation method is based on the assumption that the back-off parameters $\alpha(w_i, w_{i+1})$ change little when the system is in use and the parameters is not normalized because the re-estimation of parameters $\alpha(w_i, w_{i+1})$ is very time-consuming.

5. References

[1] S.M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 35(3): 400~401, 1987.

[2] S.F. Chen, K. Seymore, R. Rosenfeld, "Topic adaptation for language modeling using unnormalized exponential models," *ICASSP'98*, Vol. 2, pp. 681~684, 1998.

[3] M. Federico, "Efficient language model adaptation through MDI estimation." *Eurospeech'99*, Vol. 4, pp. 1583~1586, 1999.

[4] H. Masataki, Y. Sagisaka, T. Kawahara, "Task adaptation using MAP estimation in n-gram language modeling," *ICASSP'97*, pp.783~786, 1997.

[5] P.S. Rao, S. Dharanipradа, S. Roukos, "MDI adaptation of language models across corpora," *Eurospeech'97*, pp. 1979~1982, 1997.

[6] W. Reichl, "Language model adaptation using minimum discrimination information," *Eurospeech'99*, Vol. 4, pp. 1791~1794, 1999.

[7] M. Federico, "Bayesian estimation methods for n-gram language model adaptation," *ICSLP'96*, Vol. 1, pp. 180~183, 1996.

[8] R. Rosenfeld. "A maximum entropy approach to adaptive statistical language model," *Computer, Speech, and Language*, 10, 1996.

[9] L. Jin, G.-Q. Wu, F. Zheng, W.-H. Wu, "Improved strategies for intelligent sentence input method engine system," *International Symposium on Chinese Language Processing* (*SCSLP'00*), pp.247~250, 2000.

[10] K. Sasaki, H. Jiang and K. Hirose, "Rapid adaptation of n-gram language models using inter-word correlation for speech recognition," *ICSLP'00*, Vol. 4, pp. 508~512, 2000.

[11] F. Zheng, "A syllable-synchronous network search algorithm for word decoding in Chinese speech recognition," *ICASSP'99,* pp. II-601~604, March 15~19, Phoenix, 1999.