



Lexicon Adaptation with Reduced Character Error (LARCE) – A New Direction in Chinese Language Modeling

Yi-cheng Pan and Lin-shan Lee

National Taiwan University
speech@speech.ee.ntu.edu.tw

Abstract

Good language modeling relies on good predefined lexicons. For Chinese, since there are no text word boundaries and the concept of “word” is not very well defined, constructing good lexicons is difficult. In this paper, we propose lexicon adaptation with reduced character error (LARCE), which learns new word tokens based on the criterion of reduced adaptation corpus error rate. In this approach, a multi-character string is taken as a new “word” as long as it is helpful in reducing the error rate, and minimum number of new, high-quality words can be obtained. This algorithm is based on character-based consensus networks. In initial experiments on Chinese broadcast news, it is shown that LARCE not only significantly outperforms PAT-tree-based word extraction algorithms, but even outperforms manually augmented lexicons. It is believed the concept is equally useful for other character-based languages.

Index Terms: speech recognition, language model, Chinese and Japanese characters

1. Introduction

A good language model (LM) is based on a good lexicon. In Chinese language each character has its own meaning, and new words can be easily constructed by concatenating a few characters together. This leads to a very high rate of out of vocabulary (OOV) words. In addition, in printed or written Chinese texts there are no blanks serving as boundaries between words. As a result, the “word” in Chinese is not very well defined, and the segmentation of a Chinese written sentence into words is usually not unique. This leads to the high difficulties to construct good lexicons for ASR for different tasks. Moreover, a language model needs to be trained by corpora at least consistently segmented into words, but different lexicons clearly lead to differently segmented corpora. Therefore in Chinese language modeling the difficult problems of lexicon construction, word segmentation and language model training are inevitably coupled together in a complicated way [1].

In this paper, we propose a new approach: optimizing Chinese language model by lexicon adaptation based on reduced character error. The key idea is to take the ASR results on the adaptation data into account when constructing the lexicon. This is similar to the well-known discriminative training methods [2], except here our goal is to have a better lexicon rather than better acoustic models or features. In other words, a string of a few characters is taken as a “word” to be included in the lexicon as long as it is helpful to reduce the error rate, regardless of whether it looks like a word or not. Also, due to the difficulties in word segmentation, in ASR for Chinese we usually consider character error rate (CER) rather than word error rate. So the error rate considered here is also the CER. On the

other hand, including too many noisy word items not only increases the computation load, but introduces confusions. So we should try to include only those key extra items in the lexicon, but not as many items as possible. It was shown that the proposed method significantly outperforms the popularly used PAT-tree-based method [3], and even slightly outperforms the human efforts. Although all discussions here are for Chinese, it is believed the concept is equally useful for other Asian languages with similar characteristics such as Japanese.

2. Proposed Approach

2.1. Overall Picture

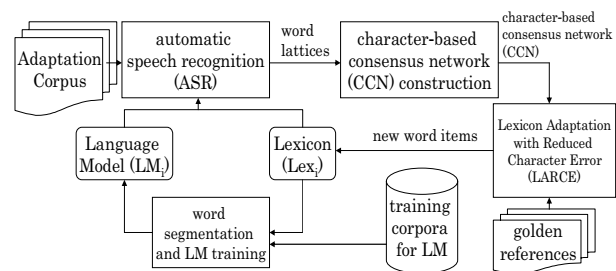


Figure 1: The flow chart of the proposed approach of optimizing Chinese language with LARCE algorithm and the CCN decoding structure.

We show the complete flow chart of the proposed method in figure 1. At the beginning we are given the adaptation spoken corpus and the golden references. Based on the baseline lexicon (Lex_0) and language model (LM_0) ($i = 0$) we perform ASR on the adaptation corpus and construct the corresponding word lattices [4]. From the word lattices, we further build the character-based consensus network (CCN) [5, 6]. We then perform the proposed algorithm of lexicon adaptation with reduced character error (LARCE) to extract new words from the CCN given the golden references. The extracted new words are then added to the current version of lexicon to re-segment the LM training corpora and the LM is in turn re-trained. This gives LM_1 and Lex_1 ($i = 1$). The whole procedure can be iterated. We will introduce the character posterior probability and character-based consensus network in Sec. 2.2 and the proposed LARCE algorithm in Sec. 2.3.

2.2. Character Posterior Probability and Character-based Consensus Network (CCN)

Consider a word W as shown in figure 2 with characters $\{c_1 c_2 c_3\}$ corresponding to the edge e starting at time τ and ending at time

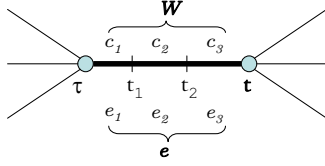


Figure 2: An edge e as word W composed of characters $c_1c_2c_3$ starts at time τ and ends at time t .

t in a word lattice. During the decoding process the boundaries between c_1 and c_2 , and c_2 and c_3 are also recorded respectively as t_1 and t_2 . The posterior probability (PP) of the edge e given the acoustic features A , $P(e|A)$, is given below [7]:

$$P(e|A) = \frac{\alpha(\tau) \cdot P(x_\tau^t|W) \cdot P_{LM}(W) \cdot \beta(t)}{\beta_{start}}, \quad (1)$$

where $\alpha(\tau)$ and $\beta(t)$ denote the forward and backward probabilities accumulated up to time τ and t as in the standard forward-backward algorithm¹, $P(x_\tau^t|W)$ is the acoustic likelihood function, $P_{LM}W$ the language model score, and β_{start} denotes the sum of all path scores in the lattice accumulated from the end time to the start time of the lattice. Equation (1) can be extended to the PP of a character of W , say c_1 with edge e_1 as in figure 2, as:

$$P(e_1|A) = \frac{\alpha(\tau) \cdot P(x_\tau^{t_1}|c_1) \cdot P_{LM}(c_1) \cdot \beta(t_1)}{\beta_{start}}. \quad (2)$$

Here we need the two new probabilities, $P_{LM}(c_1)$ and $\beta(t_1)$. Since neither is easy to estimate, we make some assumptions to obtain effective estimates. First, we assume $P_{LM}(c_1) \approx P_{LM}(W)$. Of course this is not true, the actual relation being $P_{LM}(c_1) \geq P_{LM}(W)$, since the set of events having c_1 given its history includes set of events having W given the same history (the inverse is not necessarily true). However, this is an approximation for easier implementation. Second, we assume that after c_1 there is only one path from t_1 to t : that through c_2 and c_3 . Again, this is clearly not true. There may be other paths from t_1 to t through other characters. This is again a simplifying assumption. With this assumption we have the approximation $\beta(t_1) = P(x_{t_1}^t|c_2c_3) \cdot \beta(t)$. We can now substitute these two approximate values for $P_{LM}(c_1)$ and $\beta(t_1)$ in equation (2), and the result turns out to be very simple: $P(e_1|A) \approx P(e|A)$. With similar assumptions for the character edges e_2 and e_3 , we have $P(e_2|A) \approx P(e_3|A) \approx P(e|A)$. Similar results can be observed in [6] from a different point of view.

After we obtain the PPs for each character arc in the lattice, such as $P(e_i|A)$ as mentioned above, we can perform the same procedures as previously proposed [8] to convert the lattice to a linear sequence of segments, each consisting of a set of alternatives of character hypotheses, or the character-based consensus network (CCN) [5, 6]. In CCN we actually collect the PPs for all character arc c with beginning time τ and end time t as $P([c; \tau, t]|A)$, which carries the meaning given below:

$$P([c; \tau, t]|A) = \frac{\sum_{\substack{H = w_1 \dots w_N \in \text{lattice} \\ \exists i \in \{1, \dots, N\} \\ w_i \text{ contains } [c; \tau, t]}} P(H)P(A|H)}{\sum_{\text{path } H' \in \text{lattice}} P(H')P(A|H')}, \quad (3)$$

¹We adopt the same step described in [7] to merge all nodes with identical associated times into a single node. Thus the index of the forward or backward probability mass here is specified by time rather than node number.

where H stands for a path in the lattice, or a string of words $w_1 \dots w_N$ in the lattice. $P(H)$ is the language model score of H (after proper scaling) and $P(A|H)$ is the acoustic model score. CCN was known to be very helpful in reducing CER since it minimizes the expected CER [6, 5] rather than the WER (which is minimized in word-based consensus network approach [8]) or sentence error rate (which is minimized in MAP decoding approach [9]). Given CCN, for minimized CER decoding we simply choose the characters with the highest PP in equation (3) from each segment.

2.3. Lexicon Adaptation with Reduced Character Error (LARCE)

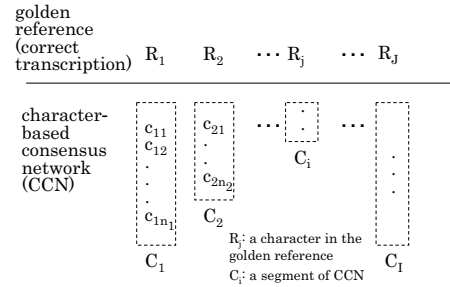


Figure 3: A character-based consensus network (CCN) and the corresponding golden reference.

In figure 3 we show a character-based consensus network (CCN) $C = \{C_i, i = 1, \dots, I\}$, where C_i is the i th segment and its corresponding golden reference (correct transcription) $R = \{R_j, j = 1, \dots, J\}$, where R_j is the j th character. In each segment C_i there are several character hypotheses sorted by their corresponding character PPs. For example, we have characters $\{c_{11} \dots c_{1n_1}\}$ in the segment C_1 . In each segment we also introduce a null character ϵ with its PP equal to 1 minus the summation of PPs for all character hypotheses in that segment.

The basic idea of the proposed LARCE approach is to enhance the PPs of those incorrectly recognized characters by adding new word items in the lexicon. We first align the CCN C with the golden reference R , so each C_i is aligned with a character $R_{align(i)}$ in R to minimize the total edit distances. $align(i)$ is an integer, $1 \leq align(i) \leq J$. If $\text{top}(C_i) \neq R_{align(i)}$, $R_{align(i)}$ will be incorrectly recognized. From equation (3), for a character $[c; \tau, t]$ to have a higher PP, it needs to be included by more paths in the lattice. The character $[c; \tau, t]$ is included by a path if the path has a word arc including the character $[c; \tau, t]$. Therefore an easy way to raise the PP of $[c; \tau, t]$ is to augment some new “words” including the character c and the adjacent characters on its right or left or both in the golden reference. For example, given the golden reference with characters $R = \{R_1, R_2, \dots, R_j, \dots, R_J\}$ if we try to increase the PP of the character R_j , we may add new “words” like $[R_j R_{j+1}]$, $[R_{j-1} R_j]$, $[R_{j-1} R_j R_{j+1}]$, for instance, into the lexicon, regardless of whether they look like a word or not. It’s possible they are OOV words. Then in decoding we may expect that there are more ways to form a path to contain the character R_j and in turn the PP of R_j may be enhanced. As a result R_j may be chosen and the character error rate may be reduced. This is the basic idea of LARCE, that is, to focus on those incorrectly recognized characters in the adaptation data and try to compose new “words” to include these characters.

Several issues should be considered in the above process of

composing new “words” as given below. (1) The added new words should not include any correctly recognized characters, because they already have the highest PP and we wish not to add any unnecessary word items. (2) For those characters in the golden reference but not even appear in the corresponding segment in CCN², we don’t add new words to include them. These characters are pruned possibly due to the mismatch of acoustic models, different speaking style or different pronunciation surface forms in those characters, for instance, and those situations won’t be improved by the lexicon and the LM. In other words, if a character is already pruned in the CCN construction process, very likely it will be pruned again in similar situations even if we have a new word covering it in the lexicon. (3) We try to add the new words with maximum length. Longer words not only reduce the number of extracted words but also are helpful for ASR [10, 11]. All these considerations are to extract only the least but the most helpful words. We wish not to introduce any unnecessary noisy word items which may lead to additional confusion. The LARCE algorithm described above is summarized in Algorithm 1. Those extracted words $\{W\}$ not existing in the current version lexicon will be the output.

Algorithm 1 LARCE Algorithm

Require: CCN C with I segments, each as a set of characters $c_{ik}, k = 1 \dots n_i$ sorted by PP, $i = 1 \dots I$
Require: Golden reference $R = \{R_j, j = 1 \dots J\}$ for C
1: Align C with R
Ensure: Each C_i is aligned with $R_{align(i)}, i = 1 \dots I$
2: $\{W\} \leftarrow \epsilon$
3: **for** $i = 0$ to I **do**
4: **if** $\text{top}(C_i) \neq \epsilon$ **then**
5: **if** $R_{align(i)} \in C_i$ **and** $\text{top}(C_i) \neq R_{align(i)}$ **then**
6: $\$begin = i$
7: **for** $ii = \$begin$ to I **do**
8: **if** $R_{align(ii)} \notin C_i$ **and** $\text{top}(C_{ii}) \neq \epsilon$ **then**
9: **goto** 12
10: **if** $\text{top}(C_{ii}) = R_{align(ii)}$ **then**
11: **goto** 12
12: $\$end = ii - 1$
13: $\{W\} = \{W\} + R_{align(\$begin)} \dots align(\$end)$
14: $i = ii$

3. Experimental Results

3.1. Baseline Lexicon, Corpora and Language Models

The baseline lexicon was automatically constructed from a 300 MB text corpus collected from local news providers from 1997 to 1999 using the widely applied PAT-tree-based word extraction method [3]. It includes 61521 words in total. The key principle of the PAT-tree-based approach is as follows. For a sequence of characters to be extracted as a word, it should (1) have high enough frequency count; (2) have high enough mutual information between component characters; (3) have large enough number of context variations on both sides; (4) not be dominated by the most frequent context among all context variations. So in general the words extracted in this way have high frequencies and clear boundaries, thus very often they have good semantic meanings. This is to imitate the way human identifies the words. Since all the above statistics of all possible character sequences in a raw corpus are combinatorially too many, we need an efficient data structure such as the PAT-tree to record and access all such information for word extraction.

With the baseline lexicon, we performed the maximum-matching algorithm [12] to segment the LM training corpora.

²That is, $R_{align(i)} \notin C_i$

Here we used three sets of LM training corpora: (1) D_1 collected in 2000, 146 MB; (2) D_2 collected in 2001, 167 MB; (3) D_{1+2} , the combination of D_1 and D_2 , 313 MB. Then we trained the three baseline language models LM_1, LM_2 and LM_{1+2} respectively using the three segmented LM training corpora.

A broadcast news corpus collected from the local radio station in Taiwan from August to September, 2001 was used as the speech corpus. So this corpus is more temporally consistent with the corpus D_2 (collected in 2001) than with D_1 . It contained 5K utterances. We separated these 5K utterances into two parts randomly: 4K as the adaptation corpus and 1K as the testing set. For the adaptation corpus, we used the manual transcriptions as the golden references to extract new words. Two algorithms were compared here: PAT-tree-based word extraction and the proposed LARCE algorithm. For PAT-tree-based word extraction, we used the manual transcriptions as the golden references to extract new words. 1530 new words were obtained and augmented into the baseline lexicon. The augmented lexicon was then used to re-segment the three LM training corpora and retrain the LMs as $LM+PAT_1, LM+PAT_2$ and $LM+PAT_{1+2}$, respectively. For the proposed LARCE algorithm, we used the three different baseline LMs, LM_1, LM_2 and LM_{1+2} , and three different new word sets containing 1967, 1568 and 1534 new words were automatically generated, respectively. Due to the limited time available, only one iteration as described in figure 1 was performed. Note that the test data (collected in 2001) is more mismatched with D_1 or LM_1 , therefore more new words were needed (1967 as compared to 1568 and 1534) to compensate for such mismatch. Similarly, 1534 new words for LM_{1+2} is less than 1568 for LM_2 alone. For both PAT-tree based and LARCE algorithm, the pronunciations of the extracted words were automatically labeled by exhaustively generating all possible pronunciations from all component characters’ possible canonical pronunciations.

3.2. Results and Discussions

The Character Accuracy (CA) results for the 1K utterances in the test set are shown in Table 1. In the upper and lower parts of Table 1, we respectively list the results for the traditional MAP decoding structure and the CCN approach mentioned in Sec. 2.2, both tested with the two different new word extraction approaches using the three different LM training corpora.

From Table 1, several observations can be made as follows. (1) The CCN decoding structure in the lower part performed better than the MAP decoding in the upper part in terms of CA in all cases. (2) The proposed LARCE approach outperformed the PAT-tree-based word extraction method in all cases (comparing the last two columns). This is because the PAT-tree method extracted new words with significant frequency counts and clear word boundaries in the raw corpus. But this didn’t guarantee improved ASR performance. LARCE, on the other hand, can compose new “words” which appear only once (or with very low frequencies) in the adaptation data, regardless of the word boundaries, while focusing on improved performance in CA. (3) For cases with higher baseline performances (with D_2 or D_{1+2}), PAT-trees resulted in smaller improvements. This is probably because some words extracted using PAT-trees are helpful for ASR but others are not. When the baseline is higher (or more useful words are already in the lexicon), there is a lower chance of extracting new words helpful to ASR using PAT-trees. But in the case of LARCE, improvements were stable even with better baselines. (4) The improvements for LARCE using CCN decoding are more significant than those using MAP

decoding, since CCN decoding is more compatible with the LARCE criterion, that is, reduced CER.

Experiments	LM training corpora	baseline	baseline +PAT	baseline +LARCE
(I) MAP decoding	D_1	74.42	74.51	75.53
	D_2	77.65	77.67	78.52
	D_{1+2}	78.16	78.18	79.02
(II) CCN decoding	D_1	75.69	75.77	76.97
	D_2	78.78	78.80	80.15
	D_{1+2}	79.28	79.29	80.48

Table 1: Character accuracy (CA) for the two decoding structures along with the two lexicon adaptation approaches : PAT-tree-based and the proposed LARCE algorithm, tested on the three different LM training corpora.

We also manually segmented the correct transcriptions to produce the manually generated new words in the next set of experiments. Due to the labor constraints, here we only manually segmented 0.5K utterances, out of which we obtained 231 new words. Using the golden references for the same 0.5K utterances (without manual segmentation), PAT-tree extracted 185 words while the proposed LARCE algorithm extracted 245, 192 and 189 respectively for LM₁, LM₂ and LM₁₊₂. The LM training corpora were also re-segmented and the LMs re-trained given all the different lexicons. The rest 4.5K utterances were then tested given the three finally trained LMs. The CA results for the 4.5K test data are shown in Table 2.

Very similar observations can be made in Table 2 as in Table 1. In addition, the most important observation here is that in all cases manually extracted words outperformed the PAT-tree-based ones, but at the same time the proposed LARCE actually outperformed the manually extracted words in all cases. In other words, some new “words” composed by LARCE were not words viewed from human, but helpful for ASR in CA; at the same time some human-extracted new words are helpless for improved CA. This result is very encouraging. It indicates a new direction to consider the lexicon adaptation: aiming for improved ASR performance rather than imitating human behavior may offer better performance than human efforts.

Experiments	LM training corpora	baseline	baseline +Manual	baseline +PAT	baseline +LARCE
(I) MAP decoding	D_1	74.56	74.58	74.57	74.65
	D_2	77.92	77.96	77.92	77.99
	D_{1+2}	78.36	78.37	78.36	78.39
(II) CCN decoding	D_1	75.83	75.85	75.84	75.93
	D_2	78.99	79.01	78.99	79.12
	D_{1+2}	79.54	79.55	79.54	79.67

Table 2: Character accuracy (CA) for the two decoding structures along with the three lexicon adaptation approaches : manual, PAT-tree-based and the proposed LARCE algorithm, tested on the three different LM training corpora.

4. Conclusion

In this paper we proposed a new approach to construct a more appropriate lexicon for ASR. It is realized using an adaptation

framework, that is, by improving the existing lexicon and the resulting LM. Contrary to the traditional concept of lexicon, we don’t care whether each new word entry looks like a word; instead we aimed at improved ASR performance in terms of character error rate reduction. This is a new direction for lexicon adaptation and language modeling and we showed very encouraging results that the lexicon adapted using LARCE even outperformed the human efforts.

5. References

- [1] Jianfeng Gao, Joshua Goodman, Mingjing Li, and Kai-Fu Lee, “Toward a unified approach to statistical language modeling for Chinese,” *TALIP*, vol. 1, no. 1, pp. 3–33, 2002.
- [2] D. Povey and C. Woodland, “Minimum phone error and i-smoothing for improved discriminative training,” in *ICASSP*, 2002.
- [3] Lee-Feng Chien, “Pat-tree-based keyword extraction for chinese information retrieval,” in *SIGIR*, 1997, pp. 50–58.
- [4] S. Ortman, H. Ney, and X. Aubert, “A word graph algorithm for large vocabulary continuous speech recognition,” *Comp. Speech Lang.*, vol. 11, pp. 43–72, 1997.
- [5] Yi-Sheng Fu, Yi-Cheng Pan, and Lin-Shan Lee, “Improved large vocabulary continuous Chinese speech recognition by character-based consensus networks,” in *ISCSLP*, 2006, pp. 422–434.
- [6] Yao Qian, Frank K. Soong, and Tan Lee, “Tone-enhanced generalized character posterior probability (GCPP) for Cantonese LVCSR,” in *ICASSP*, 2006, pp. 133–136.
- [7] F. Wessel, R. Schluter, K. Macherey, and H. Ney, “Confidence measures for large vocabulary continuous speech recognition,” *IEEE Trans. Speech and Audio Processing*, vol. 9, no. 3, pp. 288–298, Mar 2001.
- [8] L. Mangu, E. Brill, and A. Stocke, “Finding consensus among words: Lattice-based word error minimization,” in *Proc. Eurospeech*, 1999, pp. 495–498.
- [9] Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer, “A maximum likelihood approach to continuous speech recognition,” *Readings in speech recognition*, pp. 308–319, 1990.
- [10] Jianfeng Gao, Mu Li, Andi Wu, and Chang-Ning Huang, “Chinese word segmentation: A pragmatic approach,” in *MSR-TR-2004-123*, 2004.
- [11] George Saon and Mukund Padmanabhan, “Data-driven approach to designing compound words for continuous speech recognition,” *IEEE Trans. Speech and Audio Processing*, vol. 9, no. 4, pp. 327–332, May 2001.
- [12] Pak kwong Wong and Chorkin Chan, “Chinese word segmentation based on maximum matching and word binding force,” in *ICCL*, 1996, pp. 200–203.