Bilingual Continuous-Space Language Model Growing for Statistical Machine Translation

Rui Wang, Hai Zhao, Bao-Liang Lu, *Senior Member, IEEE*, Masao Utiyama, and Eiichiro Sumita

***Abstract*—Larger n-gram language models (LMs) perform better in statistical machine translation (SMT). However, the existing approaches have two main drawbacks for constructing larger LMs: 1) it is not convenient to obtain larger corpora in the same domain as the bilingual parallel corpora in SMT; 2) most of the previous studies focus on monolingual information from the target corpora only, and redundant n-grams have not been fully utilized in SMT. Nowadays, continuous-space language model (CSLM), especially neural network language model (NNLM), has been shown great improvement in the estimation accuracies of the probabilities for predicting the target words. However, most of these CSLM and NNLM approaches still consider monolingual information only or require additional corpus. In this paper, we propose a novel neural network based bilingual LM growing method. Compared to the existing approaches, the proposed method enables us to use bilingual parallel corpus for LM growing in SMT. The results show that our new method outperforms the existing approaches on both SMT performance and computational efficiency significantly.**

***Index Terms*—Continuous-space language model, language model growing (LMG), neural network language model, statistical machine translation (SMT).**

I. Introduction

NOWADAYS, many studies focus on constructing larger Back-off N-gram Language Models (BNLMs) [1], [2], [3], for a better perplexity (PPL). These larger Language Models (LMs) have been successfully applied to Statistical Machine Translation (SMT) [2] and help bring better BLEU [4], [5]. Meanwhile, larger in-domain corpora used in LM training in SMT[1] are necessary in most of the existing approaches. How to select the in-domain corpora is also a critical problem in large LM constructing and adaptation, because increasing corpora from different domains will not result in better LMs [6] or translation [7], [8], [9], [10], [11]. In addition, it is very difficult to collect an extra large corpus for some special domains such as the TED (Technology, Entertainment, Design) corpus [12] or for some rare languages. Therefore, how to improve the performance of LM without assistance of extra corpus is an important subject in SMT.

'*Language Model Growing (LMG)*' refers to adding n-grams outside the corpus together with their probabilities into the original LM. LMG is useful because it can improve LM through adding more and more useful n-grams from a small training corpus. In the past decades, various methods [13], [14], [15], [16] have been developed for adding n-grams from corpus selected by different criteria. However, none of these approaches can conduct the n-grams outside the corpus.

Recently, Continuous-Space Language Models (CSLMs), especially Neural Network Language Models (NNLMs) [17], [18], [19], [20], are actively used in SMT [21], [21], [22], [23], [24]. These models have demonstrated that CSLMs can improve BLEU scores of SMT over n-gram LMs with the same sized corpus for LM training. An attractive feature of CSLMs is that they can predict the probabilities of n-grams outside the training corpus more accurately.

Due to too high computational cost, it is difficult to use CSLMs in decoding directly. A common approach in SMT using CSLMs is a two-pass procedure, or n-best re-ranking. In this approach, the first pass uses a BNLM in decoding to produce an n-best list. Then, a CSLM is used to re-rank those n-best translations in the second pass [21], [25], [22], [23]. Another approach is based on Restricted Boltzmann Machines (RBMs) [24], instead of multi-layer neural networks [17], [18], [20]. Since the probability of an RBM can be calculated very efficiently [24], RBM-based LM can be conveniently applied to SMT decoding. However, the RBM can only be used in a small SMT task due to its high training costs.

Vaswani *et al.* propose a method for reducing the training cost of CSLM and apply it into SMT decoder [26]. However, their method is still slower than the n-gram LM. Some other studies try to implement neural network LM or translation model for SMT [27], [28], [29], [30], [31], [32], [33], [34], [35], [36]. But

Manuscript received July 27, 2014; revised November 05, 2014; accepted April 08, 2015. Date of publication April 21, 2015; date of current version May 13, 2015. The work of R. Wang, H. Zhao, and B. L. Lu was supported in part by the National Natural Science Foundation of China under Grants 60903119, 61170114, and 61272248, the National Basic Research Program of China under Grant 2013CB329401, the Science and Technology Commission of Shanghai Municipality under Grant 13511500200, the European Union Seventh Framework Program under Grant 247619, the Cai Yuanpei Program (CSC fund 201304490199, 201304490171, and the art and science interdiscipline funds of Shanghai Jiao Tong University (a study on mobilization mechanism and alerting threshold setting for online community, and media image and psychology evaluation: a computational intelligence approach under Grant 14X190040031(14JCRZ04). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiaodong He. *(Corresponding author: Hai Zhao.)*

R. Wang, H. Zhao, and B. L. Lu are with the Department of Computer Science and Engineering and Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wangrui.nlp@gmail.com; zhaohai@cs.sjtu.edu.cn; bllu@sjtu.edu.cn).

M. Utiyama and E. Sumita are with the Multilingual Translation Laboratory, National Institute of Information and Communications Technology, Kyoto 619-0289, Japan (e-mail: mutiyama@nict.go.jp; eiichiro.sumita@nict.go.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2015.2425220

[1]It is common to use larger monolingual corpus in SMT, in comparison to the small bilingual parallel corpus.

until now, the decoding speed using n-gram LM is still the state-of-the-art one.

To integrate CSLM more efficiently into decoding, some existing approaches calculate the probabilities of the n-grams before decoding and store them [37], [38], [39], [40][2]. The '*converted CSLM*' is directly used in SMT, and its decoding speed is as fast as the n-gram LM. Actually, more n-grams which are outside the training corpus can be generated by using these '*converting*' methods. Unfortunately, all of these existing approaches can only construct an LM with the similar size of the original n-gram LM.

These CSLM methods mentioned above can calculate the probabilities of the n-grams outside training corpus more accurately. However, it is difficult to decide which n-grams should be grown by using monolingual target language information for SMT, because the n-grams appearing in phrase-based SMT should be selected from the bilingual phrase table. As we know, every translation candidate phrase for phrase-based SMT decoding is from the bilingual phrase table. Therefore, additional n-grams from a larger LMs, which are outside the phrase table, will never be actually used in SMT. These n-grams are useless because they do nothing but waste computing time and storage space in LM constructing.

It is observed that the translation outputs of a phrase-based SMT system consist of phrases from either of the following cases: (a) the phrase is already included in a phrase in the phrase table, or (b) the phrase is the result of concatenating two or more phrases in the phrase table. These phases are called '*connecting phases*'. Based on this observation, the probabilities of the connecting phrases, which are not all in the training corpus, can be calculated by CSLM. Therefore, we propose a novel neural network based bilingual LM growing method for making use of the connecting phrases without assistance of extra corpus.

The rest of this paper is organized as follows. In Section II, the related work on LM growing will be introduced. A new bilingual LM growing method will be presented in Sections III and IV. In Section V, experiments will be conducted and the results will be analyzed. Section VI will summarize this work.

II. Related Work

Most of the existing LM growing methods need extra larger monolingual corpus and focus on how to select more useful n-grams from the corpus by different criteria.

Ristad and Thomas describe an algorithm for growing n-gram LM [13]. They use a greedy search for finding the individual candidate n-grams to be added to the LM by using '*Minimum Description Length (MDL)*'-based cost function. They obtain significant improvement over their baseline n-gram LM, but their baseline model performs worse as longer contexts are used according to [16]. This means that the baseline model they used [13] is not well optimized.

Niesler and Woodland present a method for backing-off from standard n-gram LMs to cluster LMs [14]. They present an approach to grow a class n-gram LM, which estimates the probability of a cluster given the possible word clusters of the context. They also use greedy search for finding the candidates to be added to the LM similar with the technique used by [13]. The difference is that they add conditional word distributions for n-gram contexts and prune away unnecessary n-grams.

Siu and Ostendorf construct n-gram LM as a tree structure and show how to combine the tree nodes in several different ways [15]. Each node of the tree represents an n-gram context and the conditional n-gram distribution for the context. Their experiments indicate that most gain can be achieved by choosing an appropriate context length separately for each word distribution, and the size of LMs can be halved with no significant loss in performance.

Siivola *et al.* present a method for estimating variable-length n-gram LM incrementally while maintaining some aspects of Kneser Ney smoothing [16], which is popular applied to several natural language processing tasks [41], [42], [43], [44], [45], [46], [47], [48]. The growing algorithm is similar to that of [14], whereas their method uses a MDL-based cost criterion. The MDL criterion is defined in a simpler manner than in the algorithm of [13], where a more compact and more theoretical criterion is developed.

It should be noted that these four kinds of methods mentioned above do not consider to grow the n-grams outside the corpus. As a result, they actually belong to LM pruning or adaptation methods from the point of view that they all construct a smaller LM from a large corpus.

As CSLM or NNLM makes it possible to calculate the probabilities of the n-gram outside the training corpus more accurately, various studies try to implement neural network LM or translation model for SMT [26], [28], [30], [31], [33], [49], [50]. However, their decoding speed is still not as fast as the n-gram LM. Because directly using CSLM in SMT or speech recognition is very time consuming, some studies focus on converting CSLM into n-gram LM.

In our previous work, we propose a method for converting CSLM into n-gram LM [37]. The probabilities of the n-grams in training corpus using CSLM are calculated, and then stored together with the n-grams in the format of n-gram LMs. The converted LM can be directly used in SMT decoding with the same speed as the original n-gram LM. The results show that this conversion method can obtain better BLEU in SMT. However, it can not generate the n-grams outside the corpus. We select this method as a baseline in this study.

Arsoy *et al.* present another CSLM converting method [39], [40]. The basic idea behind their method is that a very large LM is generated by adding all the words in the '*short-list*' of CSLM after the '*tail (end) word*' of every n-gram of the original n-gram LM[3]. The LMs are pruned into the same size of the original n-gram LM using entropy-based LM pruning method [51]. Their converted CSLM is applied to speech recognition. In fact, a larger LM has been actually grown, but this method needs to spend additional time and space on the large converted LM[4]. To our best knowledge, the method developed by Arsoy *et al.* [39], [40] is the only exiting LM growing approach that

[2]This paper is partially motivated by [38].

[3]The definition of *short-list*, *tail (end) word* and other details of CSLM will be given in Section IV.

[4]We are aware that this intermediate LM can be pruned parallel during converting, however it still costs more space than the original one depending on the threshold set for pruning.

starts from an original small corpus. We also add their method into the baselines for comparison.

In addition, all the existing LM growing methods only exploit monolingual information from corpus and do not take bilingual information into account. The performance of these grown LMs are mostly measured by PPL or word error rate in speech recognition, which is beyond the SMT topic of this paper.

III. Bilingual LM Growing

This section describes the proposed method, '*bilingual CSLM growing method' or 'connecting phrase-based CSLM growing method*'.

Following the discussion in Section I, the translation output of a phrase-based SMT system can be regarded as a concatenation of phrases in the phrase table (except unknown words). This means that an n-gram that appears in a translation output satisfies either one of the following two conditions: (a) it is included in a phrase in the phrase table or (b) it is the result of concatenating two or more phrases in the phrase table.

Based on the above observations, we propose the following procedure for constructing connecting phrases:

Step 1. All the n-grams included in the phrase table should be maintained at first;

Step 2. The connecting phrases are defined in the following way. Let w_a^b be a target language phrase starting from the a-th word ending with the b-th word, and $\beta w_a^b \gamma$ be a phrase including w_a^b as a part of it, where β and γ represent any word sequence or none. An i-gram phrase $w_1^k w_{k+1}^i$ $(1 \leq k \leq i-1)$ is a connecting phrase[5], if
   (a) w_1^k is the right (rear) part of one phrase βw_1^k in the phrase table, and
   (b) w_{k+1}^i is the left (front) part of one phrase $w_{k+1}^i \gamma$ in the phrase table.

For a 4-gram phrase '*a b c d*', it is a connecting phrase if at least one of the following conditions holds,

(a) phrases 'βa' and '*b c d* γ' are in the phrase table, or
(b) phrases 'β *a b*' and '*c d* γ' are in the phrase table, or
(c) phrases 'β *a b c*' and '$d\gamma$' are in the phrase table.

The same pipeline can be applied to other n-grams such as bigrams, trigrams and *5*-grams. After the probabilities of them are calculated using CSLM (in Section IV), the n-grams in the phrase table from Step 1 and the connecting phrases from Step 2 are combined, and the combined LM is re-normalized. Finally, the connecting phrase-based grown LM is built up.

A. Ranking the Connecting Phrases

Using connecting phrase LM growing method, the n-grams outside the corpus can be generated, and a larger LM can be constructed. Since the size of connecting phrases is too huge, which is usually more than one Terabyte (TB), it is necessary to determine the usefulness of connecting phrases which are likely to appear in SMT. More useful connecting phrases can be selected by ranking the appearing probabilities of the connecting phrases in SMT decoding. The size of the grown LM will be tuned in this way.

The translation probability $P(e|f)$ from a source phrase f to a target phrase e, can be calculated using bilingual parallel training data and found in phrase table. In decoding, the probability of a target phrase e appearing in SMT should be,

$$P_{target}(e) = \sum_f P_{source}(f) \times P(e|f), \tag{1}$$

where $P_{source}(f)$ means the appearing probability of a source phrase, which can be calculated using source language part in the parallel data. Using $P_{target}(e)$[6], a connecting phrase e with high appearing probability is selected as n-gram to add into the original LM. These n-grams are called '*grown n-grams*'.

We thus build all the connecting phrases at first, then use the appearing probabilities of the connecting phrases to decide which connecting phrases should be selected. For an i-gram connecting phrase $w_1^k w_{k+1}^i$, where w_1^k is part of βw_1^k, w_{k+1}^i is part of $w_{k+1}^i \gamma$, and the βw_1^k and $w_{k+1}^i \gamma$ are the phrases in the phrase table, the probability of the connecting phrases can be roughly estimated as,

$$P_{connecting}\left(w_1^k w_{k+1}^i\right) = \sum_{k=1}^{i-1} \left(\sum_{\beta} P_{target}\left(\beta w_1^k\right) \times \sum_{\gamma} P_{target}\left(w_{k+1}^i \gamma\right) \right), \tag{2}$$

and then the value of threshold for $P_{connecting}(w_1^k w_{k+1}^i)$ is set. It should be noted that only the connecting phrases whose appearing probabilities are higher than the threshold will be selected as the grown n-grams.

IV. Calculating the Probabilities of Grown n-grams Using CSLM

A. Standard Back-off N-gram Language Model

A BNLM predicts the probability of a word w_i given its preceding $n-1$ words $h_i = w_{i-n+1}^{i-1}$. But it will suffer from data sparseness if the context h_i does not appear in the training data. So an estimation by '*backing-off*' to models with smaller histories is necessary. In the case of the interpolated Kneser-Ney smoothing [52], the probability of w_i given h_i under a BNLM, $P_b(w_i|h_i)$, is

$$P_b(w_i|h_i) = \hat{P}_b(w_i|h_i) + \alpha(h_i) P_b(w_i|w_{i-n+2}^{i-1}), \tag{3}$$

where $\hat{P}_b(w_i|h_i)$ is a discounted probability and $\alpha(h_i)$ is the back-off weight. A BNLM is used as a background LM with a CSLM as shown in Section IV-B.

B. Continuous-Space Language Model

The main structure of a CSLM using a multi-layer neural network contains four layers: the input layer projects all words in the context h_i onto the projection layer (the first hidden layer);

[5]We are aware that connecting phrases can be applied to not only two phrases, but also three or more. However the appearing probabilities (which will be discussed in Eq. (2) of next subsection) of connecting phrases are approximately estimated. To estimate and compare probabilities of longer phrases in different lengths will lead to serious bias, and experiments also showed using more than two connecting phrases did not perform well (not shown), so only two connecting phrases are applied in this paper.

[6]This $P_{target}(e)$ hence provides more bilingual information, in comparison to only using monolingual target LMs.

the second hidden layer and the output layer achieve the non-linear probability estimation and calculate the language model probability $P(w_i|h_i)$ for the given context [18].

CSLM calculates the probabilities of all of the words in vocabulary of the corpus given the context at once. However, due to too high computational complexity of calculating the probabilities of all words, CSLM is only used to calculate the probabilities of a subset of the vocabulary. This subset is called *short-list*, which consists of the most frequent words in the vocabulary. CSLM also calculates the sum of the probabilities of all of the words outside the short-list with the help of BNLM by assigning a neuron. The probabilities of other words outside the short-list are obtained from an n-gram LM [18], [37], [53].

Let w_i, h_i be the current word and history, respectively. The CSLM with a BNLM calculates the probability of w_i given h_i, $P(w_i|h_i)$, as follows,

$$P(w_i|h_i) = \begin{cases} \frac{P_c(w_i|h_i)}{\sum_{w\in V_0} P_c(w|h_i)} P_s(h_i) & \text{if } w_i \in V_0 \\ P_b(w_i|h_i) & \text{otherwise,} \end{cases} \quad (4)$$

where V_0 is the short-list, $P_c(\cdot)$ is the probability calculated by the CSLM, $\sum_{w\in V_0} P_c(w|h_i)$ is the sum of probabilities of the neurons for all the words in the short-list, $P_b(\cdot)$ is the probability calculated by BNLM in Eq. (3), and

$$P_s(h_i) = \sum_{v\in V_0} P_b(v|h_i). \quad (5)$$

It may be regarded that CSLM redistributes the probability mass of all of the words in the short-list. This probability mass is calculated by using the n-gram LM[7] [18], [37], [53].

For our bilingual LM growing method, 5-gram BNLM and n-gram ($n = 2, 3, 4, 5$) CSLMs are built from the target language of the parallel corpus, and phrase table is built from the bilingual languages of the parallel corpus.

The probabilities of unigrams in the original BNLM will be maintained as they are. Next, the n-grams from the bilingual phrase table will be grown using the '*connecting phrases*' based method. As the number of all of the connecting phrases is very huge, the ranking method is applied to select more useful connecting phrases. The distributions of different n-grams ($n = 2, 3, 4, 5$) of the grown LMs are set as the same as the original BNLM.

The probabilities of the grown n-grams ($n = 2, 3, 4, 5$) are calculated using the 2,3,4,5-CSLMs, respectively. Namely, the grown n-grams will be the input into CSLMs, and the grown n-grams together with calculated probabilities as the output. If the tail (target) words of the grown n-grams are not in the short-list of CSLM, the $P_b(\cdot)$ in Eq. (4) will be applied to calculate the probability.

The grown n-grams ($n = 2, 3, 4, 5$) are combined together, and the probabilities and back-off weights of the n-gram LM are re-normalized using the SRILM's '*-renorm*' option [54], [55]. Finally the original BNLM and the grown LM are interpolated[8] using the default setting of SRILM. The entire process is illustrated in Fig. 1.

[7]If we do not use $P_s(h_i)$ in Eq. (4), the PPL of test data of NTCIR-9 will increase from 97.5 to 100.4. Please refer to the Section V for detail settings.

[8]The interpolation is setup for fair comparisons with Wang *et al.* [37]. and Arsoy *et al.*'s methods [39], [40], because they both use interpolation.

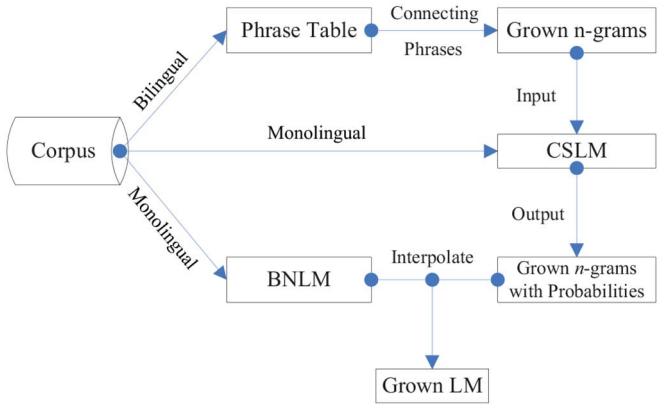


Fig. 1. Process of bilingual CSLM growing method.

C. Baseline Systems

For the baseline systems, we only re-writes the probabilities from CSLM into the BNLM in our previous work [37]. Therefore, this method can only construct a converted LM with the same size as the original BNLM. The main difference between our proposed method in this paper and our previous approach is that n-grams outside the corpus are generated firstly and the probabilities are calculated by using the same method as our previous approach. Namely, the proposed new method is the same as our previous one when no any grown n-gram is generated.

The Arsoy's method developed in [39], [40] adds all the words in the short-list after the tail words of the i-grams to construct the $(i+1)$-grams. For example, if the i-gram is '*I want*', then the $(i+1)$-gram will be '*I want*$*$', where '$*$' stands for any word in the short-list. The probabilities of the $(i+1)$-grams are calculated using the $(i+1)$-CSLM. A very large intermediate $(i+1)$-grams will have to be grown at first, and then be pruned into smaller suitable size using an entropy-based LM pruning technique modified from [51]. The $(i+2)$-grams are grown using $(i+1)$-grams recursively.

D. Computational Complexity for Growing Methods

The time complexity for CSLM [18] can be expressed as,

$$(n-1) \times P \times H + H + (H \times N) + N, \quad (6)$$

where n is the order of n-grams, P is the size of projection layer, H is the size of hidden layer, and N is the size of output layer. The original N equals to the size of the vocabulary ($|V|$). To reduce the time complexity, the short-list V_0 in Eq. (4), which is a subset of the vocabulary, is used as the output layer N. The other words outside short-list in the output layer will be calculated using background BNLM.

The method developed by Arsoy *et al.* [39], [40] is applied to 4-gram LM for speech recognition, and all of the words in the short-list are required to be added after the tail word of i-gram to construct the $(i+1)$-gram. The short-list usually includes thousands of words, so the generated $(i+1)$-grams will be thousands times larger than the i-grams before they are pruned. Because the higher order n-grams commonly contain more n-grams, so the computing cost for calculating the generated $(i+1)$-grams significantly grows as the order of n-gram increases. The low order n-gram makes their method more applicable to speech

recognition, as SMT usually requires a higher order n-gram LM. For example, 5-gram LM as common setting is used for SMT, and 4-gram LM is used for speech recognition.

In contrast, we only rewrite the n-grams from the grown LM, and calculate the same number of n-grams as the grown LM, which is at most 10 times (by experience) larger than the original one. For the baseline approach [39], [40], the size of LM is about thousands times of the original BNLM[9].

The decoding speed of the grown LM is nearly the same as the normal n-gram LM, because the neural network probabilities are encoded into the grown n-gram LM using the SRILM toolkit. As a result, the proposed method has significant advantage on computational cost. This attractive feature makes it work faster on the same corpus.

V. Experiments and Results

A. Experiment Setting

The same settings for the NTCIR-9 Chinese to English translation baseline system [56] are followed, and the only difference is to use various LMs to compare them. The Moses phrase-based SMT system is applied [57] together with GIZA++ [58] for alignment and MERT [59] for tuning on the development data. Fourteen standard SMT features are used: four translation model scores, one phrase pair number penalty score, one word penalty score, seven distortion scores, and one LM score. Each of the different LMs is used to calculate the LM score. The translation performance is measured by the case-insensitive BLEU on the tokenized test data. The tool, `mteval − v13a.pl`, is used for calculating BLEU scores[10].

The patent data for the Chinese to English patent translation subtask from the NTCIR-9 patent translation task [56][11] is used. The parallel training, development, and test data consists of 1 million (M), 2,000, and 2,000 sentences, respectively. This SMT system is called SMT1M. A subset is randomly selected from the whole train data containing 100 K sentences and all other settings the same are as the SMT1M. This SMT system is called SMT100K.

As an example, considering a phrase table built from SMT100K sentences, it consists of nearly 9.5M phrases. So it is too time and space consuming to construct the connecting phrases using all the phrases in the phrase table ($9.5\,\mathrm{M} \times 9.5\,\mathrm{M} \approx 90\,\mathrm{T}$). For SMT1M, it will take more time and space. In practice, only a small part of top useful phrases (1%-5% by experience for SMT100K and SMT1M) in the phrase table are considered by a ranking method according to Eq. (1)[12].

[9]Although all the tail words in short-list can be calculated at the same time using neurons in the output layer, their method still takes much time.

[10]It is available at http://www.itl.nist.gov/iad/mig/tests/mt/2009/

[11]We are aware that there is extra large monolingual English corpus for NTCIR-9, unlike TED, whose large extra monolingual corpus is hard to find. NTCIR-9 corpus is selected for a fair comparison among Wang *et al.* [37]'s (use NTCIR-9 corpus), Arsoy *et al.* [39], [40]'s and our methods. The experiments on TED corpus will be shown in Section V-E, and the experiments using additional monolingual corpus will be shown in Section V-F.

[12]We also empirically compare the proposed ranking method with the phrase table pruning method in Johanson *et al.* [60], and the ranking method shows better performance. Thus we choose the ranking method at last.

TABLE I
Coverage Rates of Words in Short-List

Sizes of Short-list	Coverage Rates (%)
1K (training)	75.20
4K (training)	86.28
8K (training)	92.89
16K (training)	93.82
24K (training)	94.69
8K (test)	91.08

Following SRILM [54], [55], a 5-gram BNLM is trained with interpolated Kneser-Ney smoothing method using the 1M/100K sentences or 42M/4M words without cutoff. These LMs are called BNLM42M and BNLM4M.

The 2,3,4,5-CSLMs are trained on the same 1M/100K sentences using the CSLM toolkit [53]. The settings for the CSLMs are: projection layer (first hidden layer) of dimension 256 for each word, (second) hidden layer of dimension 384 and output layer (short-list) of dimension 8192, which are recommended in the CSLM toolkit and our previous work [37]. These CSLMs are called CSLM42M and CSLM4M.

In this paper, around 42M words are used as the corpus, including 456 K words as vocabulary, and 8 K words, which covers 92.89% of words in the training corpus, as short-list for both our method and baseline methods. Arsoy *et al.* use around 55M words as the corpus, 84 K words as vocabulary, and 20 K words as short-list. The sizes of our corpus and short-list are similar with theirs in [39], [40]. However, our vocabulary is much larger than theirs, because the whole vocabulary must be used for SMT decoding, compared with only a small vocabulary used for speech recognition. The ratio of $|V_0|/|V|$ affect how much the time complexity will be reduced for the output layer. With a small $|V|$ (84 K) in Arsoy *et al.*'s [39], [40], 76% time cost could be saved as 20 K is chosen for the short-list size. For our method, $|V|$ being 456 K and 8 K being short-list, 98% time cost is saved.

The coverage rate, C_r, of short-list is defined as:

$$C_r = \frac{N_{short-list}}{N_{corpus}} \times 100\%, \tag{7}$$

where $N_{short-list}$ indicates the number of words hit in short-list and N_{corpus} indicates the number of words in corpus. The size of short-list is selected according to the corpus by experiments before CSLM is constructed. The results in Table I show that the coverage rates become saturated after short-list size is larger than 8 K. It should be noted that these 1% and 2% of coverage rate differences determine which LM, CSLM or BNLM, is more likely used to calculate the probabilities of n-grams. However, it will lead to a big gap of computational cost. For a 8 K sized short-list, nearly 93% of the n-grams in the training data will be calculated using CSLM and 7% using BNLM. The time complexity for CSLM is approximately linear with the size of output layer (short-list). If the short-list is 20 K as that in Arsoy's method, it will take nearly 2.5 times of computing time for training and converting CSLM. Therefore, the 8 K short-list is adopted for both methods. In comparison with 20 K short-list, 8 K makes little loss on the coverage rate, but brings about much higher efficiency.

TABLE II
PROPORTIONS OF n-GRAMS COVERED BY CSLM AND/OR BNLM

	CSLM	BNLM	short-list	back-off	Proportions (%)
A	In	Out	In	Yes	65.49
B	Out	In	Out	No	2.50
C	In	In	In	No	25.59
D	Out	Out	Out	Yes	6.42

TABLE III
PERFORMANCE OF THE GROWN LMS IN SMT100K

LMs	n-grams	PPL	BLEU-s	BLEU-i	ALH
BNLM4M	10.3M	144.8	27.48	27.48	2.57
CSLM4M	N/A	**126.6**	N/A	28.03	N/A
Wang4M	10.3M	140.5	27.69	27.72	2.57
Arsoy4M-1	33.3M	135.1	27.70	27.90	2.68
Arsoy4M-2	53.3M	134.7	27.72	27.81	2.69
Arsoy4M-3	68.4M	134.3	27.82	27.71	2.71
Arsoy4M-4	89.5M	134.2	27.83	27.92	2.72
Arsoy4M-5	107.9M	134.1	27.74	27.83	2.73
Bil4M-1	33.5M	134.8	27.94+	27.97	2.74
Bil4M-2	49.4M	134.1	28.05++	28.06+	2.78
Bil4M-3	65.6M	133.2	28.00+	27.99++	2.80
Bil4M-4	90.1M	132.5	27.99+	**28.07**+	2.82
Bil4M-5	110.3M	132.2	**28.10**++	28.02+	**2.83**

The proportions of n-grams in the test data covered by CSLM (8 K as short-list) and/or BNLM[13] are also counted. For each n-gram, there are four situations shown in Table II.

It should be noted that the common settings of CSLM in our previous work [37], Arsoy *et al.*'s approach and the proposed bilingual growing method are all the same for fair comparison with the same 2.70 GHz CPUs in this paper. Code for the bilingual CSLM growing is available online[14].

B. SMT Results

Experiments are firstly conducted on SMT100K. That is, CSLM4M and BNLM4M are used for LMs growing, and then the grown LMs are applied to SMT100K. The results are shown in Table III. Then, the experiments are conducted on SMT1M. That is, CSLM42M and BNLM42M are used for LMs growing, and then the grown LMs are applied to SMT1M. The results are shown in Table IV. At last, the grown LMs from CSLM4M and BNLM4M are applied to SMT1M to study whether the grown LMs built from a small corpus can outperform the original BNLMs built from larger corpus. In other words, CSLM4M and BNLM4M are used for LMs growing, and then the grown LMs are applied to SMT1M. The results are shown in Table V.

The results of the LM experiments in SMT are divided into five groups: the original BNLMs (BNLM4M/42M), CSLMs in decoding (CSLM4/42M, mainly following the setting of [50], [61]), our previous converting method [37], Arsoy *et al.* [39], [40]'s growing, and neural network based bilingual growing methods. For our previous method [37] which is referred to Wang4M/42M, the probabilities are only re-written from CSLM into the BNLM. Therefore, only an LM with the same size can be conducted. For our new bilingual LM growing method, 5

[13]If the n-grams are not covered by BNLM, BNLM will refer to lower-order probabilities with the adjusted weights using back-off.

[14]it is available at http://bcmi.sjtu.edu.cn/wangrui/program/blmg.zip

TABLE IV
PERFORMANCE OF THE GROWN LMS IN SMT1M

LMs	n-grams	PPL	BLEU-s	BLEU-i	ALH
BNLM42M	73.9M	108.8	32.19	32.19	3.03
CSLM42M	N/A	**97.5**	N/A	33.18	N/A
Wang42M	73.9M	104.4	32.60	32.62	3.03
Arsoy42M-1	217.6M	103.3	32.55	32.75	3.14
Arsoy42M-2	323.8M	103.1	32.61	32.64	3.18
Arsoy42M-3	458.5M	103.0	32.39	32.71	3.20
Arsoy42M-4	565.6M	102.8	32.67	32.51	3.21
Arsoy42M-5	712.2M	102.5	32.49	32.60	3.22
Bil42M-1	223.5M	101.9	32.81+	33.02+	3.20
Bil42M-2	343.6M	101.0	32.92+	33.11++	3.24
Bil42M-3	464.5M	100.6	33.08++	**33.25**++	3.26
Bil42M-4	571.0M	100.3	**33.15**++	33.12++	3.28
Bil42M-5	705.5M	100.1	33.11++	33.24++	**3.31**

TABLE V
PERFORMANCE OF THE GROWN LMS (BIL4M) IN SMT1M

LMs	n-grams	PPL	BLEU-s	BLEU-i	ALH
BNLM42M	73.9M	108.8	32.19	32.19	**3.03**
Wang42M	73.9M	**104.4**	**32.60**	**32.62**	**3.03**
Bil4M-1	33.5M	135.2	29.45	29.98	2.74
Bil4M-2	49.4M	134.1	29.47	30.62	2.78
Bil4M-3	65.6M	133.2	29.46	30.49	2.80
Bil4M-4	90.1M	132.5	29.57	30.31	2.82
Bil4M-5	110.3M	132.2	29.60	30.56	2.83

Bilingual grown LMs for every SMT system (**Bil**4M/42M-1 to 5[15]) are conducted in increasing sizes, and the largest grown LM is around 10 times larger than the original one by the number of n-grams. For the method in [39], [40], 5 grown LMs (Arsoy4M/42M-1 to 5) for each SMT system are also conducted in increasing sizes. Entropy based method is adopted to prune them into similar sizes as the grown LMs using our method (Bil4M/42M-1 to 5).

Our previous converted LM, Arsoy grown LMs and bilingual grown LMs are interpolated with the original BNLMs by using default setting of SRILM[16]. To reduce the randomness of MERT, two methods are used for tuning the weights of different SMT features, and two BLEU scores are obtained corresponding to these two methods. **BLEU-s** indicates that the **same** weights of the baseline (BNLM4M/42M) features are used for all the SMT systems. **BLEU-i** indicates that the MERT is run **independently** by three times and the average BLEU scores are taken. The trends of PPL and BLEU-s are illustrated in Figs. 2 and 3, respectively.

The paired bootstrap re-sampling test [62][17] is performed. 2000 samples are used for each significance test. The marks at the right of the BLEU scores indicate whether the LMs are significantly better/worse than Arsoy's grown LMs with the same

[15]We firstly select the top connecting phrases, which are around 10 times larger than the n-gram (phrases) in the original BNLM, and then choose the top 20%, 40%, 60%, 80% and 100% of them to construct the Bil42M-1 to Bil42M-5. The IDs after the Bil42M indicate the sizes of grown LMs (in ascending order). The distribution of n-grams in different orders ($n = 2, 3, 4, 5$) is the same as the original BNLM.

[16]Our previous method [37] uses the development data to tune the weights of interpolation. In this paper, the default weight 0.5 is used as the interpolation weights for fair comparison.

[17]The implementation of our system follows http://www.ark.cs.cmu.edu/MT

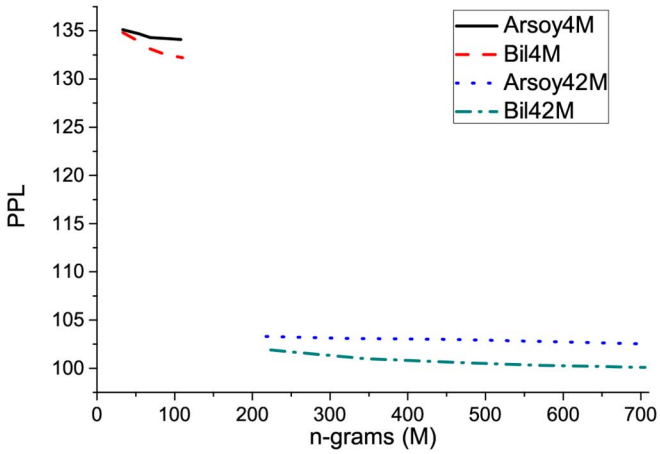


Fig. 2. Trend of PPL as the LMs grow (Lower is better).

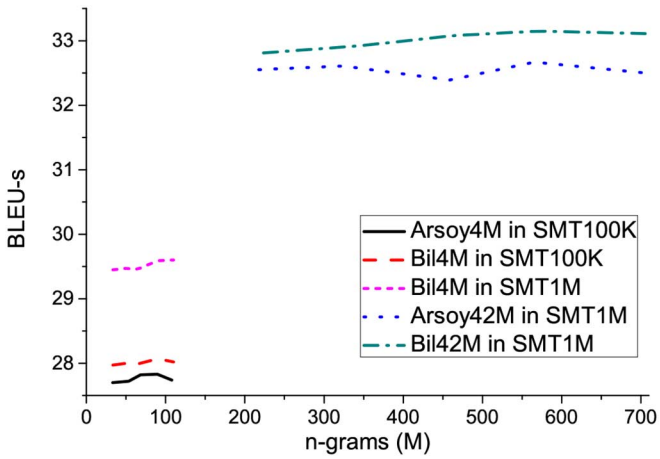


Fig. 3. Trend of BLEU as the LMs grow (Higher is better).

TABLE VI
PERFORMANCE OF THE GROWN LMS AND CSLM IN RE-RANKING

LMs	n-grams	PPL	BLEU (Re-ranking)
BNLM42M	73.9M	108.8	32.19
CSLM42M	N/A	**97.5**	32.46
Bil42M-1	223.5M	101.9	32.31
Bil42M-2	343.6M	101.0	32.44
Bil42M-3	464.5M	100.6	32.37
Bil42M-4	571.0M	100.3	**32.51**
Bil42M-5	705.5M	100.1	32.28

IDs ('++/−−': significantly better/worse at significance level $\alpha = 0.01$, '+/−': $\alpha = 0.05$).

Comparison are also conducted on re-ranking for CSLM and our bilingual grown LMs. CSLM42M and Bil42M are used to re-rank the 1000-best lists of SMT1M. That is, the BNLM scores in the 1000-best lists are replaced with the CSLM42M/Bil42M scores, and then the global scores are re-ranked. For CSLM re-ranking in Table VI, the PPL column indicates that the PPLs are calculated using CSLM, and the BLEU(Re-ranking) column indicates that the feature weight of CSLM42M/Bil42M is tuned using Z-MERT [63]. The results are presented in Table VI.

From the results shown in Tables III, IV, V, and VI and Figs. 2 and 3, we can obtain the following observations:

TABLE VII
CONSTRUCTING CONNECTING PHRASES TIME

SMT Systems	Constructing Time (sec.)
SMT100K	11.3
SMT1M	17.9

(a) Nearly all the bilingual grown LMs outperform the original BNLM and our previous converted LM on the PPL and BLEU in SMT. These indicate that our bilingual LM growing method can give better probability estimation for LM and better performance for SMT. Compared with the CSLM in re-ranking and decoding methods, the bilingual grown LMs obtain higher PPLs, but similar BLEUs.

(b) As the sizes of grown LM increase, the PPLs on the test data always decrease and the BLEU scores trend to increase. Bil42M-1 keeps the top 20% ranked connecting phrases, which are the most useful connecting phrases, and therefore it performs similar as Bil42M-5, which contains all the selected connecting phrases.

(c) Compared with the grown LMs in [39], [40], our grown LMs obtain better PPL and significantly better BLEU with the similar size. Furthermore, the improvements on the PPL and BLEU obtained by their method become saturated much more quickly than ours, as the LMs grow.

(d) The grown LMs (Bil4M) in SMT1M perform much better than in SMT100K, but much worse than the BNLM42M with the similar size in SMT1M. This indicates that the grown LMs built from small corpus can indeed improve the performance on PPL and BLEU. But they do not perform well compared to the LMs originally built from large corpus[18].

C. Efficiency Comparison

In this subsection, the efficiency of our new bilingual LM growing method is investigated.

1) Efficiency for Constructing Connecting Phrases: The average time for constructing 1M trigrams connecting phrases for SMT100K and SMT1M is shown in Table VII. The results indicate that the constructing time is much less than the growing time in Table VIII for the same 1M trigrams.

2) Growing Time: The growing time of Arsoy's method and the proposed bilingual growing method is evaluated. The 1M trigrams are used as the input n-grams for CSLMs (SMT100K for CSLM4M and SMT1M for CSLM42M) for both methods. The time of LMs growing is recorded. For the growing step, Arsoy's method will generate much more n-grams and then prune it into smaller ones[19]. The proposed method only produces the same size n-grams. The growing time used by these two methods is presented in Table VIII.

From Table VIII, we can see that Arosy's growing method takes much more time (nearly 40 times) than ours. The Arsoy's method grows all the possible n-grams ended with the words in

[18]Given a large extra monolingual corpora (in-domain and out-of-domian), our connecting phrase method will be suitable for LM domain adaptation. Please refer to Section V-F for details.

[19]The pruning actually takes a lot of time, but we do not take it into account for fair comparison of growing time.

TABLE VIII
GROWING TIME

CSLMs	Arosy	Our
CSLM4M	24 hours 10 minutes	37 minutes
CSLM42M	24 hours 24 minutes	41 minutes

TABLE IX
DECODING TIME

LMs	Decoding Time (sec.)
BNLM42M	**15.3**
CSLM42M	186.5
NPLM42M	50.4
NPLM42M (normalized)	456.8
Bil42M-3	16.5

the short-list (usually thousands times of the original one) and then prunes them into smaller ones. Although all the tail words in the short-list can be calculated at the same time using neurons in the output layer in CSLM, Arsoy's method still takes much more time than ours. For the proposed method, which n-grams to be grown are decided according to the appearing probabilities in Eq. (2) before they are put into CSLM. The growing time for the proposed method is approximately linear with the size of input n-grams. In practice, a grown LM which is around 8 times larger than the original one has the best performance in SMT as shown in Tables III, IV and V.

3) Decoding Time: Vaswani *et al.* [26] propose several techniques (such as NCE training algorithm) to improve the efficiency of using NPLM in SMT. Meanwhile our proposed growing method stores the probabilities of the CSLM into BNLM format to improve the efficiency of using CSLM in SMT. In this subsection, we evaluate the decoding efficiency of the BNLM, CSLM, NPLM[20] and our bilingual grown LM with all the same settings in SMT decoding. The 2,000 English reference sentences of the NTCIR-9 test data are used for evaluation. The log-probabilities of every sentence of the test data are calculated using different LMs for SMT decoding stimulation. The decoding time is shown in Table IX.

From Table IX, the following conclusions can be reached:

(a) The decoding time using Bil42M-3 and BNLM are quite close. This indicates that the proposed converting method can run as fast as the BNLM.
(b) The decoding time using CSLM is much slower than the BNLM and Bil42M-3. This demonstrates the advantage of decoding speed of the proposed method.
(c) The decoding time using the NPLM (without normalization) is much faster than the CSLM and much slower than the BNLM/Bil42M-3. This also shows the efficiency advantage of our proposed method.

D. Function of Connecting Phrase

The above results show that the proposed LM growing method performs better in terms of both PPL and BLEU, because more useful connecting phrases have been added to the LM. This subsection will show how these connecting phrases perform in SMT in detail.

[20]The numbers of hidden layers are set the same as the CSLM and other settings follow the default setting of NPLM toolkit [26].

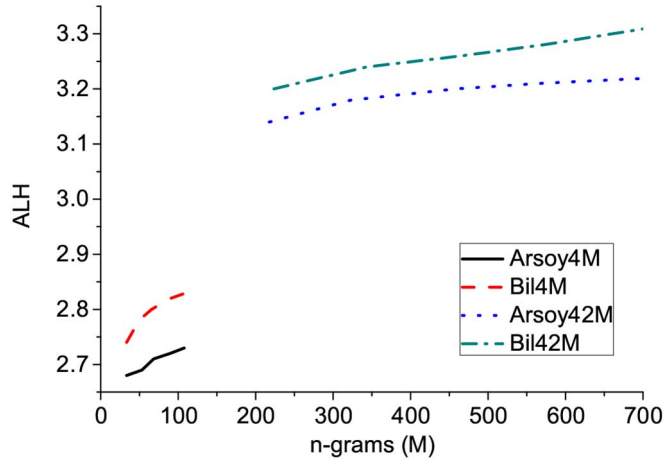


Fig. 4. Trend of ALH in SMT decoding (Longer is better).

Given each sentence from the test data, the probability of every target word is calculated using the grown LMs, to simulate the performance of the LMs in SMT decoding. Then the ratio of the different n-grams used for each grown LMs on all the test data is counted.

The results of the i-grams hit in SMT decoding are shown in Tables III, IV and V. The last column is the Average Length of the n-grams Hit (ALH) in SMT decoding for different LMs using the following function,

$$ALH = \sum_{i=1}^{5} P_{i-gram} \times i, \tag{8}$$

where P_{i-gram} means the percentage of the i-grams hit in SMT decoding, and it is illustrated in Fig. 4.

There exist positive correlations among the ALH, PPL and BLEUs. The ALHs of bilingual grown LMs (Bil42/4M) are longer than those of Arsoy's grown LMs in similar sizes (Arsoy42/4M).

As we know, the LM refers to the probabilities of $(i-1)$-grams together with the adjusted weights using back-off, if no any corresponding i-gram is hit. The statistics above indicate that more high-order n-grams are hit using the proposed grown LMs in SMT in comparison to the Arsoy's grown LMs. In another word, less back-off is applied to the proposed connecting phrase-based grown LMs in SMT decoding.

E. Experiments on TED Corpus

As TED corpus is in special domain, where large extra monolingual corpora are hard to be found. In this subsection, the SMT experiments on TED corpora are conducted by using the proposed LM growing method.

The baselines of the IWSLT 2014 evaluation campaign[21] are followed and only a few modifications are made such as the LM toolkits and n-gram order for constructing LMs. The French (FR) to English (EN) and Chinese (CN) to English language pairs are chosen. The data sets, dev2010 and test2010, are selected as development data and evaluation data, respectively. The statistics on TED parallel data used for setting up the baselines are described in Table X.

[21]It is available at https://wit3.fbk.eu

TABLE X
STATISTICS ON TED PARALLEL DATA

FR-EN	Sentences
training	186.8K
dev2010	0.9K
test2010	1.6K
CN-EN	**Sentences**
training	186.8K
dev2010	0.9K
test2010	1.6K

TABLE XI
FR-EN TED EXPERIMENTS

LMs	*n*-grams	PPL	BLEU-s
BNLM	8.2M	90.0	32.30
Wang	8.2M	85.4	32.67
Bil-1	27.1M	83.2	32.87
Bil-2	52.4M	82.6	33.05
Bil-3	79.8M	**81.8**	**33.21**

TABLE XII
CN-EN TED EXPERIMENTS

LMs	*n*-grams	PPL	BLEU-s
BNLM	7.8M	87.1	12.41
Wang	7.8M	85.3	12.73
Bil-1	23.1M	79.2	12.92
Bil-2	49.7M	78.3	13.16
Bil-3	73.4M	**77.6**	**13.24**

The same LM growing method is applied to TED corpora as on NTCIR corpora. That is, the bilingual data is used to grow the LMs, and then the monolingual grown LMs are integrated into SMT system. The results of TED experiments are divided into three groups: baseline method using BNLM (BNLN)[22], our previous method using converted CSLM (Wang) [37], and our proposed bilingual grown LMs (Bil1-1,2,3) on both language pairs. The results are shown in Tables XI and XII.

Tables XI and XII demonstrate that the proposed growing method can improve the performances of LMs in both PPL and BLEU for different corpora and language pairs.

F. Experiments on Additional Monolingual Corpora

So far, the concerned LMs are limited to the target side of bilingual corpora. In this subsection, the CSLMs using additional monolingual corpora are constructed and compared with our bilingual grown LMs without using additional monolingual corpora.

We conduct experiments on both in-domain and out-of-domain corpora. For the in-domain NTCIR-9 patent experiments, the 2005 US patent English data set distributed in the NTCIR-8 patent translation task [64] is used as the additional monolingual corpus, which consists of around 5M sentences[23]. For the out-of-domain TED experiments, the NIST English OpenMT06 data set[24], which consists of around 400 K sentences, is used following the same setting of [65]. TED data set belongs to a specific domain, so it is not easy to obtain in-domain additional monolingual corpus in fact. The additional US patent data and NIST data are added to the original NTCIR-9 and TED data, respectively, to construct the large monolingual corpora. These corpora are called Corpus-US-patent and Corpus-NIST, respectively. The statistics on these corpora are shown in Table XIII.

TABLE XIII
STATISTICS OF ADDITIONAL MONOLINGUAL CORPORA

Corpora	Sentences	Words
Corpus-NTCIR	1.0M	42.3M
Corpus-US-patent	5.9M	202.9M
Corpus-TED	186.8K	2.7M
Corpus-NIST	591.7K	15.2M

TABLE XIV
NTCIR EXPERIMENTS WITH ADDITIONAL MONOLINGUAL CORPUS

LMs	*n*-grams	PPL	BLEU-s
BNLM-NTCIR	73.9M	108.8	32.19
BNLM-US-patent	312.4M	95.7	32.76
Converted CSLM-US-patent-(a)	312.4M	**91.3**	**33.24**
Converted CSLM-US-patent-(b)	312.4M	97.5	32.85
Bil42M-2 (NTCIR)	343.6M	101.0	32.92

TABLE XV
TED (CN-EN) EXPERIMENTS WITH ADDITIONAL MONOLINGUAL CORPUS

LMs	*n*-grams	PPL	BLEU-s
BNLM-TED	7.8M	87.1	12.41
BNLM-NIST	26.5M	111.2	8.03
Converted CSLM-NIST-(a)	26.5M	102.1	10.87
Converted CSLM-NIST-(b)	26.5M	96.3	11.32
Bil-1 (TED)	23.1M	**79.2**	**12.92**

The LM experimental settings are conducted in two ways: (a) both CSLMs and BNLMs are built from the large corpora (Corpus-US-patent/NIST), and the large BNLMs are converted using large CSLMs. These converted CSLMS are called Converted CSLM-US-patent/NIST-(a); (b) only the BNLMs are built from the large corpora, and the large BNLMs are converted using small CSLMs built from NTCIR-9/TED corpora. These converted CSLMS are called Converted CSLM-US-patent/NIST-(b).

The SMT experimental settings follow Section V-A, except for using different LMs. That is, the same CN-EN phrase table and other models used in Section V-A are applied, and LMs are built from different size corpora. The bilingual grown LMs (Bil42M-2 (NTCIR) and Bil-1 (TED)), with the similar size as the corresponding Converted CSLMs built from large additional corpora, are selected for fair comparison.

The results are shown in Tables XIV and XV. For the NTCIR patent experiments, the Converted CSLM-US-patent using additional monolingual in-domain patent corpus perform better than our bilingual grown LM. For the TED SMT experiments, our bilingual grown LMs perform better than the Converted

[22]Our CN-EN baseline is a little better than the baseline (BLEU $= 11.21$) in IWSLT 2014. For the FR-EN language pair, only the EN-FR baseline is shown (BLEU $= 29.44$) in IWSLT 2014. Please refer to https://wit3.fbk.eu/score.php?release=2014-01 for details.

[23]The original corpus consists of 25M sentences, and we choose a part of these sentences randomly.

[24]It is available at http://www.itl.nist.gov/iad/mig/tests/mt/2006/. The data mainly consists of news and blog texts.

TABLE XVI
HIERARCHICAL PHRASE-BASED SMT ON CN-EN NTCIR

LMs	*n*-grams	PPL	BLEU-s
BNLM	73.9M	108.8	33.14
Bil42M-1	223.5M	101.9	33.38
Bil42M-3	464.5M	100.6	33.64
Bil42M-5	705.5M	**100.1**	**33.71**

TABLE XVII
SYNTAX-BASED SMT ON CN-EN NTCIR

LMs	*n*-grams	PPL	BLEU-s
BNLM	73.9M	108.8	29.03
Bil42M-1	223.5M	101.9	29.39
Bil42M-3	464.5M	100.6	**29.51**
Bil42M-5	705.5M	**100.1**	29.42

TABLE XVIII
HIERARCHICAL PHRASE-BASED SMT ON FR-EN TED

LMs	*n*-grams	PPL	BLEU-s
BNLM	8.2M	90.0	31.79
Bil-1	27.1M	83.2	32.03
Bil-2	52.4M	82.6	**32.35**
Bil-3	79.8M	**81.8**	32.12

TABLE XIX
SYNTAX-BASED SMT ON FR-EN TED

LMs	*n*-grams	PPL	BLEU-s
BNLM	8.2M	90.0	28.64
Bil-1	27.1M	83.2	28.87
Bil-2	52.4M	82.6	29.10
Bil-3	79.8M	**81.8**	**29.21**

CSLM-NIST using the out-of-domain additional monolingual corpus. These results suggest that our bilingual LM growing method is useful for corpus adaptation.

G. Experiments on Hierarchical Phrase-based and Syntax-based SMT

The experiments of applying our bilingual grown CSLMs to hierarchical phrase-based and syntax-based SMT are also conducted. Namely, the bilingual grown CSLMs are firstly constructed using phrase table, and then applied to hierarchical phrase-based or syntax-based SMT. The same settings for the NTCIR-9 hierarchical translation baseline system [56] are followed for hierarchical phrase-based SMT, and WAT-2014 String-to-Tree translation baseline system[25] [66] for syntax-based SMT.

Experiments are conducted on NTCIR-9 Chinese-to-English and IWSLT-2014 TED French-to-English corpora, and results are shown in Tables XIV, XVII, XVIII and XIX.

The results in Tables XVI, XVII, XVIII and XIX demonstrate that the proposed LM growing method can improve the BLEU for different translation models.

[25]To accelerate the speed of parsing and training, the maximum length limit of sentences is set a little shorter than the baseline.

VI. CONCLUSIONS

A novel proposed LM growing method in this paper has two attractive features. First, it constructs a large efficient LM using neural network. The pre-computed CSLM probabilities inside/outside the corpus are stored in BNLM format, and therefore the grown LMs can perform as precisely as CSLM and run as fast as the BNLM. Second, it takes the phrase-table into consideration, and makes the grown LM obtain bilingual information for SMT. The key points of our method are to construct and select the connecting phrases, which are likely to appear in SMT decoding but outside phrase-table. The ranking functions are carefully designed for constructing the connecting phrases precisely and efficiently.

Various metrics are applied to evaluate our new method, and the experimental results show that the proposed method significantly outperforms the existing LM converting/growing methods in SMT performance. Both the growing time and decoding time are also significantly reduced. From the experimental results, we can see that the proposed method is promising and can be applied to LM adaptation for additional monolingual out-of-domain corpus.

ACKNOWLEDGMENT

We appreciate the helpful discussions with Dr. Isao Goto, Prof. Sabine Ploux, Zhongye Jia, anonymous reviewers and editors for many invaluable comments and suggestions to improve this paper.

REFERENCES

[1] R. Jonson, "Generating statistical language models from interpretation grammars in dialogue systems," in *Proc. 11th Conf. Eur. Assoc. Comput. Linguist.*, Trento, Italy, Apr. 2006, pp. 57–65, Assoc. for Comput. Linguist..

[2] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, "Large language models in machine translation," in *Proc. Joint Conf. Empir. Meth. Nat. Lang. Process. Comput. Nat. Lang. Learn.*, Prague, Czech Republic, Jun. 2007, pp. 858–867.

[3] K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn, "Scalable modified kneser-ney language model estimation," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguist. (Vol. 2: Short Papers)*, Sofia, Bulgaria, Aug. 2013, pp. 690–696.

[4] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. for Comput. Linguist.*, Philadelphia, PA, Jun. 2002, pp. 311–318.

[5] J. Zhang and H. Zhao, "Improving function word alignment with frequency and syntactic information," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2211–2217.

[6] J. R. Bellegarda, "Statistical language model adaptation: Review and perspectives," *Speech Commun.*, vol. 42, no. 1, pp. 93–108, 2004.

[7] P. Koehn and J. Schroeder, "Experiments in domain adaptation for statistical machine translation," in *Proc. 2nd Workshop Statist. Mach. Translat.*, 2007, pp. 224–227.

[8] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proc. Conf. Empir. Meth. Nat. Lang. Process.*, Edinburgh, U.K., Jul. 2011, pp. 355–362.

[9] J. Zhang and C. Zong, "Learning a phrase-based translation model from monolingual data with application to domain adaptation," in *Proc. 51st Annu. Meeting Assoc. for Comput. Linguist.*, Sofia, Bulgaria, Aug. 2013, pp. 1425–1434.

[10] H. Zhao, M. Utiyama, E. Sumita, and B. L. Lu, "An empirical study on word segmentation for chinese machine translation," *Comput. Linguist. Intell. Text Process.*, ser. Lecture Notes in Computer Science, vol. 7817, pp. 248–263, 2013.

[11] K. Duh, G. Neubig, K. Sudoh, and H. Tsukada, "Adaptation data selection using neural language models: Experiments in machine translation," in *Proc. 51st Annu. Meet. Assoc. for Comput. Linguist.*, Sofia, Bulgaria, Aug. 2013, pp. 678–683.

[12] M. Cettolo, C. Girardi, and M. Federico, "Wit^3: Web inventory of transcribed and translated talks," in *Proc. 16th Conf. Eur. Assoc. Mach. Translat.*, Trento, Italy, May 2012, pp. 261–268.

[13] E. S. Ristad and R. G. Thomas, "New techniques for context modeling," in *Proc. 33rd Annu. Meeting Assoc. Comput. Linguist.*, Cambridge, MA, USA, 1995, pp. 220–227.

[14] T. Niesler and P. Woodland, "A variable-length category-based n-gram language model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1996, vol. 1, pp. 164–167.

[15] M. Siu and M. Ostendorf, "Variable n-grams and extensions for conversational speech language modeling," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 1, pp. 63–75, Jan. 2000.

[16] V. Siivola, T. Hirsimki, and S. Virpioja, "On growing and pruning Kneser-Ney smoothed n-gram models," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 5, pp. 1617–1624, 2007.

[17] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Mar. 2003.

[18] H. Schwenk, "Continuous space language models," *Comput. Speech Lang.*, vol. 21, no. 3, pp. 492–518, 2007.

[19] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, Makuhari, Japan, 2010, pp. 1045–1048.

[20] H.-S. Le, I. Oparin, A. Allauzen, J. Gauvain, and F. Yvon, "Structured output layer neural network language model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Prague, Czech Republic, May 2011, pp. 5524–5527.

[21] H. Schwenk, D. Dchelotte, and J.-L. Gauvain, "Continuous space language models for statistical machine translation," in *Proc. COLING/ ACL*, Sydney, Australia, Jul. 2006, pp. 723–730.

[22] H. Schwenk, A. Rousseau, and M. Attik, "Large, pruned or continuous space language models on a gpu for statistical machine translation," in *Proc. NAACL-HLT 2012Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, Montreal, QC, Canada, Jun. 2012, pp. 11–19, ser. WLM '12.

[23] L. H. Son, A. Allauzen, and F. Yvon, "Continuous space translation models with neural networks," in *Proc. Conf. North Amer. Chap/ Assoc. for Comput. Linguist.: Human Lang. Technol.*, Montreal, QC, Canada, Jun. 2012, pp. 39–48.

[24] J. Niehues and A. Waibel, "Continuous space language models using restricted Boltzmann machines," in *Proc. Int. Workshop Spoken Lang. Translat.*, Hong Kong, 2012, pp. 311–318.

[25] L. H. Son, A. Allauzen, G. Wisniewski, and F. Yvon, "Training continuous space language models: Some practical issues," in *Proc. Conf. Empir. Meth. Natural Lang. Process.*, Cambridge, MA, USA, Oct. 2010, pp. 778–788, ser. EMNLP '10.

[26] A. Vaswani, Y. Zhao, V. Fossum, and D. Chiang, "Decoding with large-scale neural language models improves translation," in *Proc. 2013Conf. Empir. Meth. Nat. Lang. Process.*, Seattle, WA, USA, Oct. 2013, pp. 1387–1392.

[27] J. Gao, X. He, W.-t. Yih, and L. Deng, "Learning continuous phrase representations for translation modeling," in *Proc. 52nd Annu. Meeting Assoc. for Comput. Linguist.*, Baltimore, MD, USA, Jun. 2014, pp. 699–709.

[28] J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, and J. Makhoul, "Fast and robust neural network joint models for statistical machine translation," in *Proc. 52nd Annu. Meeting Assoc. for Comput. Linguist.*, Baltimore, MD, USA, Jun. 2014, pp. 1370–1380.

[29] J. Zhang, M. Utiyama, E. Sumita, and H. Zhao, "Learning hierarchical translation spans," in *Proc. Conf. Empir. Meth. Nat. Lang. Process.*, Doha, Qatar, Oct. 2014, pp. 183–188.

[30] M. Auli, M. Galley, C. Quirk, and G. Zweig, "Joint language and translation modeling with recurrent neural networks," in *Proc. Conf. Empir. Meth. Nat. Lang. Process.*, Seattle, WA, USA, Oct. 2013, pp. 1044–1054.

[31] L. Liu, T. Watanabe, E. Sumita, and T. Zhao, "Additive neural networks for statistical machine translation," in *Proc. 51st Annu. Meeting Assoc. for Comput. Linguist.*, Sofia, Bulgaria, Aug. 2013, pp. 791–801.

[32] M. Sundermeyer, T. Alkhouli, J. Wuebker, and H. Ney, "Translation modeling with bidirectional recurrent neural networks," in *Proc. Conf. Empir. Meth. Nat. Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 14–25.

[33] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proc. Conf. Empir. Meth. Nat. Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

[34] W. Y. Zou, R. Socher, D. Cer, and C. D. Manning, "Bilingual word embeddings for phrase-based machine translation," in *Proc. Conf. Empir. Meth. Nat. Lang. Process.*, Seattle, WA, USA, Oct. 2013, pp. 1393–1398.

[35] S. Lauly, H. Larochelle, M. Khapra, B. Ravindran, V. C. Raykar, and A. Saha, "An autoencoder approach to learning bilingual word representations," *Adv. Neural Inf. Process. Syst.*, pp. 1853–1861, 2014.

[36] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proc. 2013 Conf. Empir. Meth. Nat. Lang. Process.*, Seattle, WA, USA, Oct. 2013, pp. 1700–1709.

[37] R. Wang, M. Utiyama, I. Goto, E. Sumita, H. Zhao, and B. L. Lu, "Converting continuous-space language models into n-gram language models for statistical machine translation," in *Proc. Conf. Empir. Meth. Nat. Lang. Process.*, Seattle, WA, USA, Oct. 2013, pp. 845–850.

[38] R. Wang, H. Zhao, B. L. Lu, M. Utiyama, and E. Sumita, "Neural network based bilingual language model growing for statistical machine translation," in *Proc. Conf. Empir. Meth. Nat. Lang. Process.*, Doha, Qatar, Oct. 2014, pp. 189–195.

[39] E. Arsoy, S. F. Chen, B. Ramabhadran, and A. Sethy, "Converting neural network language models into back-off language models for efficient decoding in automatic speech recognition," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 8242–8246.

[40] E. Arsoy, S. F. Chen, B. Ramabhadran, and A. Sethy, "Converting neural network language models into back-off language models for efficient decoding in automatic speech recognition," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 22, no. 1, pp. 184–192, Jan. 2014.

[41] X. Zhang, H. Zhao, and C. Hui, "A machine learning approach to convert CCGbank to Penn treebank," in *Proc. 24th Int. Conf. Comput. Linguist.*, Mumbai, India, Dec. 2012, pp. 535–542.

[42] Q. Xu and H. Zhao, "Using deep linguistic features for finding deceptive opinion spam," in *Proc. 24th Int. Conf. Comput. Linguist.*, Mumbai, India, Dec. 2012, pp. 1341–1350.

[43] X. Ma and H. Zhao, "Fourth-order dependency parsing," in *Proc. 24th Int. Conf. Comput. Linguist.*, Mumbai, India, Dec. 2012, pp. 785–796.

[44] Z. Jia and H. Zhao, "A joint graph model for pinyin-to-chinese conversion with typo correction," in *Proc. 52nd Annu. Meeting Assoc. for Comput. Linguist.*, Baltimore, MD, USA, Jun. 2014, pp. 1512–1523.

[45] X. Wang, H. Zhao, and B. L. Lu, "A meta-top-down method for large-scale hierarchical classification," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 3, pp. 500–513, Mar. 2014.

[46] H. Zhao, "Character-level dependencies in chinese: Usefulness and learning," in *Proc. 12th Conf. Eur. Chap. ACL (EACL'09)*, Athens, Greece, Mar. 2009, pp. 879–887.

[47] H. Zhao, Y. Song, C. Kit, and G. Zhou, "Cross language dependency parsing using a bilingual lexicon," in *Proc. Joint Conf. 47th Annu. Meeting ACL and 4th Int. Joint Conf. Nat. Lang. Process. AFNLP*, Suntec, Singapore, Aug. 2009, pp. 55–63.

[48] H. Zhao, W. Chen, and C. Kit, "Semantic dependency parsing of nombank and propbank: An efficient integrated approach via a large-scale feature selection," in *Proc. Conf. Empir. Meth. Nat. Lang. Process.*, Singapore, Aug. 2009, pp. 30–39.

[49] S. Liu, N. Yang, M. Li, and M. Zhou, "A recursive recurrent neural network for statistical machine translation," in *Proc. 52nd Annu. Meeting Assoc. for Comput. Linguist.*, Baltimore, MD, USA, Jun. 2014, pp. 1491–1500.

[50] H. Schwenk, "Continuous space translation models for phrase-based statistical machine translation," in *Proc. 24th Int. Conf. Comput. Linguist.*, Mumbai, India, Dec. 2012, pp. 1071–1080.

[51] A. Stolcke, "Entropy-based pruning of backoff language models," in *Proc. DARPA Broadcast News Transcript. Understand. Workshop*, Lansdowne, VA, USA, 1998, pp. 270–274.

[52] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Comput. Speech Lang.*, vol. 13, no. 4, pp. 359–393, 1999.

[53] H. Schwenk, "Continuous-space language models for statistical machine translation," *Prague Bull. Math. l Linguist.*, pp. 137–146, 2010.

[54] A. Stolcke, "Srilm-an extensible language modeling toolkit," in *Proc. Int. Conf. Spoken Lang. Process.*, Seattle, WA, USA, Nov. 2002, pp. 257–286.

[55] A. Stolcke, J. Zheng, W. Wang, and V. Abrash, "SRILM at sixteen: Update and outlook," in *Proc. IEEE Autom. Speech Recogn. Understand. Workshop*, Waikoloa, HI, USA, Dec. 2011.

[56] I. Goto, B. Lu, K. P. Chow, E. Sumita, and B. K. Tsou, "Overview of the patent machine translation task at the NTCIR-9 workshop," in *Proc. NTCIR-9 Workshop Meeting*, Tokyo, Japan, Dec. 2011, pp. 559–578.

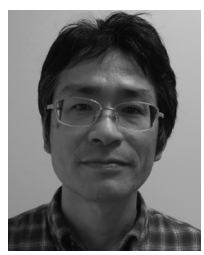
[57] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *Proc. 45th Annu. Meeting Assoc. for Comput. Linguist.*, Prague, Czech Republic, Jun. 2007, pp. 177–180.
[58] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Comput. Linguist.*, vol. 29, no. 1, pp. 19–51, Mar. 2003.
[59] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. 41st Annu. Meeting Assoc. for Comput. Linguist.*, Sapporo, Japan, Jul. 2003, pp. 160–167.
[60] H. Johnson, J. Martin, G. Foster, and R. Kuhn, "Improving translation quality by discarding most of the phrasetable," in *Proc. Joint Conf. Empir. Meth. Nat. Lang. Process. and Comput. Nat. Lang. Learn.*, Prague, Czech Republic, Jun. 2007, pp. 967–975.
[61] H. Schwenk, A. Rousseau, and M. Attik, "Large, pruned or continuous space language models on a gpu for statistical machine translation," in *Proc. NAACL-HLT Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT*, Montreal, QC, Canada, Jun. 2012, pp. 11–19.
[62] P. Koehn, , D. Lin and D. Wu, Eds., "Statistical significance tests for machine translation evaluation," in *Proc. Conf. Empir. Meth. Nat. Lang. Process.*. Barcelona, Spain: , Jul. 2004, pp. 388–395.
[63] O. F. Zaidan, "Z-MERT: A fully configurable open source tool for minimum error rate training of machine translation systems," *Prague Bull. Math. Linguist.*, vol. 91, pp. 79–88, 2009.
[64] A. Fujii, M. Utiyama, M. Yamamoto, and T. Utsuro, "Overview of the patent translation task at the NTCIR-8 workshop," in *Proc. 8th NTCIR Workshop Meeting Eval. Inf. Access Technol.: Inf. Retrieval, Question Answering and Cross-Lingual Inf. Access*, Tokyo, Japan, Jun. 2010, pp. 293–302.
[65] X. Wang, M. Utiyama, A. Finch, and E. Sumita, "Empirical study of unsupervised chinese word segmentation methods for smt on large-scale corpora," in *Proc. 52nd Annu. Meeting Assoc. for Comput. Linguist.*, Baltimore, MD, USA, Jun. 2014, pp. 752–758.
[66] T. Nakazawa, H. Mino, I. Goto, S. Kurohashi, and E. Sumita, "Overview of the 1st Workshop on Asian Translation," in *Proc. 1st Workshop Asian Translat.*, Tokyo, Japan, Oct. 2014, pp. 1–19.

Rui Wang received the B.S. degree in computer science from the Harbin Institute of Technology, China, in 2009 and the M.S. degree in computer science from the Chinese Academy of Sciences, China, in 2012. He is currently a Ph.D. candidate at Shanghai Jiao Tong University from 2012. He was an internship research fellow at National Institute of Information and Communications Technology, Japan, in 2013 and Joint Ph.D. in the National Center for Scientific Research, France, in 2014. His research interests include machine learning, natural language processing, and machine translation.

Hai Zhao received the B.S. degree in sensor and instrument and the M.S. degree in control theory and engineering from Yanshan University, and the Ph.D. degree in computer science from Shanghai Jiao Tong University, China. He is currently an Associate Professor at Shanghai Jiao Tong University. He was a Postdoctoral Research Fellow at the City University of Hong Kong from 2006 to 2009. His research interests include machine learning, natural language processing, data mining and artificial intelligence.

Bao-Liang Lu (M'94–SM'01) received the B.S. degree in instrument and control engineering from the Qingdao University of Science and Technology, Qingdao, China, in 1982, the M.S. degree in computer science and technology from Northwestern Polytechnical University, Xian, China, in 1989, and the Dr. Eng. degree in electrical engineering from Kyoto University, Kyoto, Japan, in 1994.

He was with the Qingdao University of Science and Technology from 1982 to 1986. From 1994 to 1999, he was a Frontier Researcher with the Bio-Mimetic Control Research Center, Institute of Physical and Chemical Research (RIKEN), Nagoya, Japan, and a Research Scientist with the RIKEN Brain Science Institute, Wako, Japan, from 1999 to 2002. Since 2002, he has been a Full Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, where he has been an Adjunct Professor with the Laboratory for Computational Biology, Shanghai Center for Systems Biomedicine, since 2005. His current research interests include brain-like computing, neural network, machine learning, computer vision, bioinformatics, brain–computer interface, and affective computing. Prof. Lu was the President of the Asia Pacific Neural Network Assembly (APNNA) and the General Chair of the 18th International Conference on Neural Information Processing in 2011. He is currently an Associate Editor of Neural Networks and a Board Member of APNNA.



Masao Utiyama completed his doctoral dissertation at the University of Tsukuba in 1997.

He is a Senior Researcher of the National Institute of Information and Communications Technology, Japan. His main research field is machine translation.

Eiichiro Sumita received the Bachelor and Master degree in computer science from The University of Electro-Communications, Japan, in 1980 and 1982 and the Ph.D. degree in engineering from Kyoto University, Japan in 1999. He is currently Director of Multilingual Translation Laboratory of National Institute of Information and Communication Technology from 2006. He was with the Advanced Telecomunications Research Institute International from 1992 to 2009 and IBM Research-Tokyo from 1980 to 1991. His research interests include Machine Translation and e-Learning.