The ICT Statistical Machine Translation Systems for the IWSLT 2009

Haitao Mi, Yang Liu, Tian Xia, Xinyan Xiao, Yang Feng, Jun Xie
Hao Xiong, Zhaopeng Tu, Daqi Zheng, Yajuan Lu, Qun Liu

Key Laboratory of Intelligent Information Processing, Institute of Computing Technology
Chinese Academy of Sciences
No.6 Kexueyuan South Road, Haidian District
P.O. Box 2704, Beijing, China, 100080
{htmi, yliu, xiatian, xiaoxinyan, fengyang, xiejun, lvyajuan, liuqun}@ict.ac.cn

Abstract

This paper describes the ICT Statistical Machine Translation systems that used in the evaluation campaign of the International Workshop on Spoken Language Translation (IWSLT) 2009. For this year's evaluation, we participated in the Challenge Task (Chinese-English and English-Chinese) and BTEC Task (Chinese-English). And we mainly focus on one new method to improve single system's translation quality. Specifically, we developed a sentence-similarity based development set selection technique. For each task, we finally submitted the single system who got the maximum BLEU scores on the selected development set. The four single translation systems are based on different techniques: a linguistically syntax-based system, two formally syntax-based systems and a phrase-based system. Typically, we didn't use any rescoring or system combination techniques in this year's evaluation.

1. Introduction

This paper describes the statistical machine translation systems of Institute of Computing Technology, Chinese Academy of Sciences(ICT-CAS) for the evaluation campaign of the International Workshop on Spoken Language Translation (IWSLT) 2009.

For this year's evaluation, our group participated in three tasks:

1. BTEC task, Chinese-English direction;
2. Challenge task, Chinese-English direction;
3. Challenge task, English-Chinese direction.

For each task of IWSLT 2009, the final submition is one of the four single systems who achieved a maximum BLEU score on development set. The four different systems are listed below:

1. Silenus, a linguistically syntax-based system that converts source-forest into target-string with tree-to-string rules acquired from packed forests;
2. Bruin, a formally syntax-based system that implements a maximum entropy based reordering model on BTG rules;
3. Chiero, a formally syntax-based system that employs hierarchical phrases;
4. Moses, a phrase-based open source system [1].

This paper is organized as follows: Section 2 gives an overview of our four SMT systems, Section 3 describes data preparation. In Section 4, we will report the experiments and results. Finally, Section 5 gives conclusions.

2. Single Systems Overview

2.1. Silenus

Silenus [1, 2] is a linguistically syntax-based SMT system, which employs packed forests in both training and decoding rather than *single-best* trees used in conventional tree-to-string model [3, 4].

Informally, a packed parse forest, or *forest* in short, is a compact representation of all the derivations (i.e., parse trees) for a given sentence under a context-free grammar [5]. Silenus searches for the best derivation (a sequence of tree-to-string rules) d^* that converts a source tree T in the forest into a target string s among all possible derivations D::

$$d^* = \arg\max_{d \in D} P(d|T) \quad (1)$$

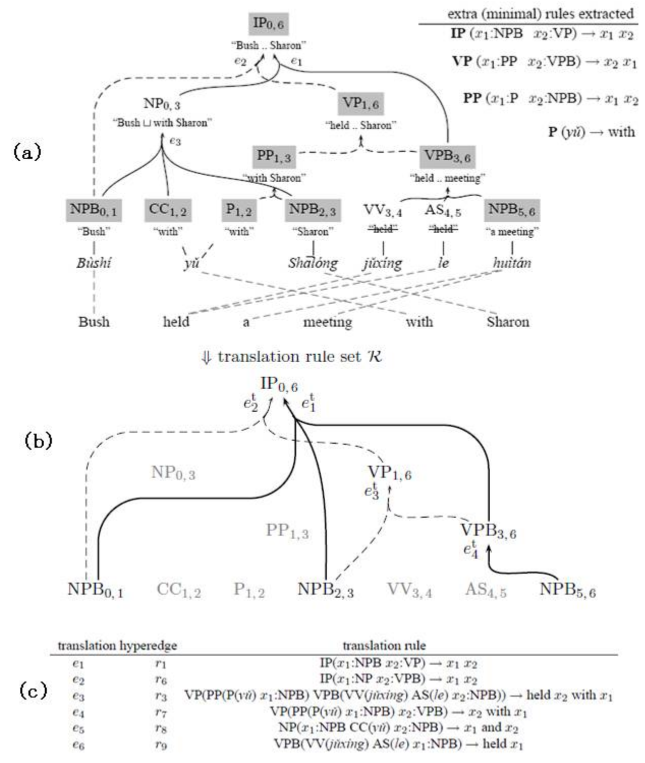
We extract rules from word-aligned bilingual corpus with source forests F (Figure 1 (a)) in two steps:

(1) frontier set computation (where to cut), and

(2) fragmentation (how to cut).

Basically, we compute the frontier set according to GHKM [6] algorithm. We highlight the nodes in frontier set by gray shades in Figure 1(a).

[1] http://www.statmt.org./moses/



translation hyperedge		translation rule
e_1	r_1	IP(x_1:NPB x_2:VP) → x_1 x_2
e_2	r_6	IP(x_1:NP x_2:VPB) → x_1 x_2
e_3	r_3	VP(PP(P(*yǔ*) x_1:NPB) VPB(VV(*jǔxíng*) AS(*le*) x_2:NPB)) → held x_2 with x_1
e_4	r_7	VP(PP(P(*yǔ*) x_1:NPB) x_2:VPB) → x_2 with x_1
e_5	r_8	NP(x_1:NPB CC(*yǔ*) x_2:NPB) → x_1 and x_2
e_6	r_9	VPB(VV(*jǔxíng*) AS(*le*) x_1:NPB) → held x_1

Figure 1: Forest-based Rule Extraction and Translation

The fragmentation step is formalized by a breadth-first search (BFS) algorithm. The basic idea is to visit each frontier node v, and keep a queue $open$ of growing fragments rooted at v. We keep expanding incomplete fragments from $open$, and extract a rule if a complete fragment is found . Each fragment in $open$ is associated with a list of *expansion sites* ($exps$ being the subset of leaf nodes of the current fragment that are *not* in the frontier set (recall that expansion stops at frontier-set nodes). So each initial fragment along hyperedge h is associated with

$$exps = tails(h) \setminus fs.$$

A fragment is complete if its expansion sites is empty, otherwise we pop one expansion node v' to grow and spin-off new fragments by following hyperedges of v', adding new expansion sites, until all active fragments are complete and $open$ queue is empty.

Five probabilities will be computed for each rule r, where $lhs(r)$ denotes the left-hand-side of r, and $rhs(r)$ denotes the right-hand-side of r, while the $root(lhs(r)$ denotes the root node of the tree-fragment $lhs(r)$.

$$P(f(r|lhs(r)) = \frac{f(r)}{\sum_{lhs(r')=lhs(r)} f(r')} \tag{2}$$

$$P(f(r|rhs(r))) = \frac{f(r)}{\sum_{rhs(r')=rhs(r)} f(r')} \tag{3}$$

$$P(lex(lhs(r)|rhs(r))) \tag{4}$$

$$P(lex(rhs(r)|lhs(r))) \tag{5}$$

$$\begin{aligned} &P(f(r|root(lhs(r)))) \\ &= \frac{f(r)}{\sum_{root(lhs(r'))=root(lhs(r))} f(r')} \end{aligned} \tag{6}$$

When computing the lexical translation probabilities described in [7], we only take the terminals into account. If there are no terminals, we set the feature value to 1.

At the decoding time, we first parse the input sentences into forests. and then we convert the parse forest into a *translation forest*(Figure 1(b)) by pattern-matching. Finally, Silenus searches for the best derivation on the translation forest and outputs the target string.

Beside the features we computed in rule extraction procedure, the additional features used in decoding step are listed here:

- The number of rules in the derivation;
- The number of words in the target translation;
- The language model score for the target translation;
- The source side parsing probability of the tree traversed by the deviation.

The decoder performs two tasks on the translation forest: 1-best search with integrated language model (LM), and k-best search with LM to be used in minimum error rate training. Both tasks can be done efficiently by forest-based algorithms based on k-best parsing [8].

For 1-best search, we use the *cube pruning* technique [9, 10] which approximately intersects the translation forest with the LM. Basically, cube pruning works bottom up in a forest, keeping at most k +LM items at each node, and uses the best-first expansion idea from the Algorithm 2 of [8] to speed up the computation.

For k-best search after getting 1-best derivation, we use the lazy Algorithm 3 of [8] that works backwards from the root node, incrementally computing the second, third, through the kth best alternatives. However, this time we work on a finer-grained forest, called *translation+LM* forest, resulting from the intersection of the translation forest and the LM, with its nodes being the +LM items during cube pruning. Although this new forest is prohibitively large, Algorithm 3 is very efficient with minimal overhead on top of 1-best.

For more details, please refer to [1] and [2].

2.2. Bruin

Bruin is a formally syntax-based SMT system, which implements the maximum entropy based reordering model on BTG [11] rules. This model considers the reorder as a problem of classification, where the Maximum Entropy model is introduced.

To complete the decoding procedure, three BTG rules are used to derive the translation:

$$A \overset{[\,]}{\rightarrow} (A^1, A^2) \tag{7}$$

$$A \overset{\langle\,\rangle}{\rightarrow} (A^1, A^2) \tag{8}$$

$$A \rightarrow (x, y) \tag{9}$$

The lexical rule (3) is used to translate source phrase y into target phrase x and generate a block A. The merging rules (1) and (2) are used to merge two consecutive blocks into a single larger block in the straight or inverted order.

Three essential elements must be illustrated in Bruin. The first one is a stochastic BTG, whose rules are weighted using different features in the log-linear form. The second is a MaxEnt-based reordering model predicting the orders between neighbor blocks, whose features are automatically learned from bilingual training data. The last one is a CKY-style chart-based decoder with beam search which is similar to that of Wu [11].

To construct a stochastic BTG, we calculate rule probabilities by the log-linear model. For the two merging rules *straight* and *inverted*, applying them on two consecutive blocks A^1 and A^2 is assigned a probability $Pr^m(A)$

$$Pr^m(A) = \Omega^{\lambda_\Omega} \cdot \triangle^{\lambda_{LM}}_{p_{LM}(A^1,A^2)} \tag{10}$$

where the Ω is the reordering score of block A^1 and A^2, which is calculated by the MaxEnt-based reordering model, λ_Ω is its weight. The $\triangle_{p_{LM}(A^1,A^2)}$ is the increment of the language model score of the two blocks according to their final order, λ_{LM} is its weight.

For the lexical rule, applying it is assigned a probability $Pr^l(A)$:

$$\begin{aligned} Pr^l(A) \quad = \quad & p(x|y)^{\lambda_1} \cdot p(y|x)^{\lambda_2} \cdot p_{lex}(x|y)^{\lambda_3} \\ & \cdot p_{lex}(y|x)^{\lambda_4} \cdot exp(1)^{\lambda_5} \cdot exp(|x|)^{\lambda_6} \\ & \cdot p_{LM}^{\lambda_{LM}}(x) \end{aligned} \tag{11}$$

where $p(\cdot)$ are the phrase translation probabilities in both directions, $p_{lex}(\cdot)$ are the lexical translation probabilities in both directions, and $exp(1)$ and $exp(|x|)$ are the phrase penalty and word penalty, respectively.

The feature weights λs are tuned to maximize the BLEU score on the development set, using minimum-error-rate training [12].

The MaxEnt-based Reordering Model (MRM) is defined on the two consecutive blocks A^1 and A^2 together with their order $o \in \{straight, inverted\}$ according to the maximum entropy framework.

$$\Omega = p_\theta(o|A^1, A^2) = \frac{exp(\sum_i \theta_i h_i(o, A^1, A^2))}{\sum_o exp(\sum_i \theta_i h_i(o, A^1, A^2))} \tag{12}$$

where the functions $h_i \in \{0, 1\}$ are model features and the θ_i are the weights.

The decoder is built upon the CKY chart-based algorithm. We use cube pruning technology to speed up the decoding.

For more details, please refer to [13].

2.3. Chiero

Chiero is a re-implementation of the state-of-the-art hierarchical phrase-based model [9].

This model can be formalized as a synchronous context-free grammar, which is automatically acquired from word-aligned parallel data without any syntactic information.

$$X \rightarrow < \gamma, \alpha, \sim > \tag{13}$$

Where X is a non-terminal, γ, α are strings of terminals and non-terminals, and $\sim$ is one-to-one correspondence between the non-terminal in γ, α.

Our work faithfully followed Chiang's [9] work. The only exception is the condition for terminating cube pruning. Chiang's [9] implementation quits upon considering the next item if its score falls outside the beam by more than ϵ. However we found that a large number of items will often be enumerated under this condition in our experiments. So we further limit the number of items popped from the heap.

Additionally, we also have conducted different experiments on two kinds of k-best lists: the true or counterfeit k-best lists. For the former method, each hypothesis must store the information of recombination items when we search for the single-best translation. Then we use the best-first expansion idea from the Algorithm 2 of Liang Huang[8] to generate the k-best lists. By contrast, for the latter method, we just discard the recombination items at single-best searching time. Experimental results show that the true k-best lists can get better results even with a less beam size than the counterfeit ones. The main reason may lie in the stable feature weights tuned on the true k-best lists.

2.4. Moses

Moses is a phrase-based model. It is an open source system [2] and uses beam-search to reduce the searching space. We will use the default settings for this model in this year's evaluation.

3. Data Preparation

For this year's evaluation, we only use the data provided by the organizer. We first used the Chinese lexical anal-

[2] http://www.statmt.org./moses/

ysis system ICTCLAS for splitting Chinese characters into words and a rule-based tokenizer for tokenizing English sentences. Then,we convert all alphanumeric characters to their 2-byte representation. Finally, we ran GIZA++ and used the "grow-diagfinal" heuristic to get many-to-many word alignments.

We used the SRI Language Modeling Toolkit [14] to train the Chinese/English 5-gram language model with Kneser-Ney smoothing on the Chinese/English side of the training corpus respectively.

Regarding to Silenus, we used the Chinese parser of [15] and English parser of [16] to parse the source and target side of the bilingual corpus into packed forests respectively. Then we pruned the forests with the marginal probability-based inside-outside algorithm [17] with a pruning threshold $p_e = 3$. At the decoding time, we use a large pruning threshold $p_d = 12$ to generate the packed forest.

3.1. Development Set Selection

Our development set for this year's evaluation is selected automatically from all the development sentences according to the n-gram similarity, which is calculated against the current test set sentences.

Our method works as follows: First, we gather every n-gram(up to 10) in the test set into a map W. and assign a score S_w for each n-gram w in W, which is calculated as

$$S_w(w) = n \cdot count(w) \tag{14}$$

where $count(w)$ is the number of occurrence of w in test set. Then, we assign a sentence score S_s to each candidate sentence s in development set, which is calculated as:

$$S_s(s) = \frac{\sum_{w \in W} S_w(w) \cdot count_s(w)}{length(s)} \tag{15}$$

where the $count_s(w)$ is the number of occurrence of w in s, and the $length(s)$ is the number of words in s. Finally, we choose the top k sentences with different thresholds as our new development set.

4. Experiments

4.1. Results on IWSLT08

We first test our development set selection method on the test set of IWSLT08. The running single system in this section is Chiero. The thresholds are integers from 1 to 5.

The final results are shown on Figure2. The bottom line is the BLEU scores when we tune feature weights on IWSLT07, while the top line is the performances when we tune weights on test set of IWSLT08. Then the results of our dev selection method are shown on the middle line, whose points are associated with the sentence numbers in each dev set. So we can conclude that our selection method improves the performance of our single system.

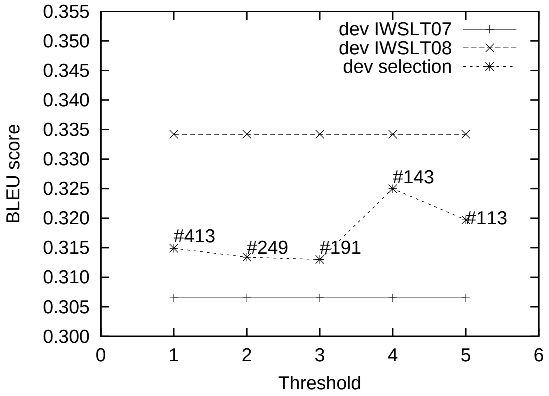


Figure 2: The BLEU scores and sentence # of dev selection with different thresholds.

4.2. Results on IWSLT09

Table 1 gives the BLEU scores (case-insensitive, with punctuations) of our four single systems achieved on the dev sets we selected, where "BTEC_CE" denotes Chinese-English direction of BTEC task, "CT_CE" denotes Chinese-English direction of challenge task, and "CT_EC" denotes English-Chinese direction of challenge task.

For each task of this year's evaluation, the final primary system is the system, who achieves the MAX BLEU score on dev set. So we chose Moses for BTEC_CE task, Chiero for CT_CE task and Silenus for CT_EC task accordingly.

Table 1: *The BLEU scores of four single systems on dev set.*

System \ Task	BTEC_CE	CT_CE	CT_EC
Bruin	0.4204	0.3521	0.4623
Chiero	0.4359	0.3732	0.4369
Moses	0.4683	0.3645	0.4734
Silenus	0.4489	0.3649	0.4775

The final BLEU scores (case-sensitive, with punctuations) of each primary system on test set are shown in Tabler̃eftb:test, where "CRR" denotes correct recognition results and "ASR.20" denotes using 20-best ASR results. When we run decoders on 20-best results, we simply decode them one by one and then output the result with highest score.

From Table 2, although Silenus achieves an higher BLEU score of 0.3886 and wins the third place on CT_EC CRR task, the correspondent score on ASR.20 task is very low, which is only 0.2901. The main reason lies in the different parsing quality on two set. With too much noise in ASR results, the parser failed to generate good forest, which will hurt the performance inevitably. The other reason maybe that we use too much ASR results (20-best) without using any rescoring technic. The simple way of choosing the highest among all translation results of 20-best results may lower the output

Table 2: *The BLEU scores of each primary single system on test set.*

Task	Input	System	BLEU
BTEC	CRR	Moses	0.3563
CT_CE	CRR ASR.20	Chiero	0.3078 0.2859
CT_EC	CRR ASR.20	Silenus	0.3886 0.2901

quality. Another thing we can conclude is that our Silenus perform better on English-Chinese direction than Chinese-English mainly due to the higher parsing quality on English.

5. Conclusions

In this paper, we describes the ICT statistical machine translation systems for the evaluation campaign of IWSLT 2009. We first used a selection method to construct a development set for each task. Then we run all the single systems on each dev set. Finally, we choose the system with maximum BLEU score as our primary system for each task. Since we didn't use any rescoring or system combination techniques for the final submitions, we got a relatively lower rank. We also concluded that our linguistically syntax-based system performs better on English-Chinese direction than Chinese-English due to the higher parsing quality on English.

6. Acknowledgements

This authors were supported by National Science Foundation of China, Contracts 60736014 and 0873167, and 863 State Key Project No. 2006AA10108.

7. References

[1] H. Mi, L. Huang, and Q. Liu, "Forest-based translation," in *Proceedings of ACL*, 2008.

[2] H. Mi and L. Huang, "Forest-based translation rule extraction," in *Proceedings of EMNLP*, Honolulu, Hawaii, October 2008, pp. 206–214.

[3] Y. Liu, Q. Liu, and S. Lin, "Tree-to-string alignment template for statistical machine translation," in *Proceedings of COLING-ACL*, Sydney, Australia, July 2006, pp. 609–616.

[4] Y. Liu, Y. Huang, Q. Liu, and S. Lin, "Forest-to-string statistical translation rules," in *Proceedings of ACL*, Prague, Czech Republic, June 2007, pp. 704–711.

[5] S. Billot and B. Lang, "The structure of shared forests in ambiguous parsing," in *Proceedings of ACL '89*, 1989, pp. 143–151.

[6] M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer, "Scalable inference and training of context-rich syntactic translation models," in *Proceedings of COLING-ACL*, Sydney, Australia, July 2006, pp. 961–968.

[7] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proceedings of HLT-NAACL*, Edmonton, Canada, May 2003, pp. 127–133.

[8] L. Huang and D. Chiang, "Better k-best parsing," in *Proceedings of Ninth International Workshop on Parsing Technologies (IWPT-2005)*, Vancouver, Canada, 2005.

[9] D. Chiang, "Hierarchical phrase-based translation," *Comput. Linguist.*, vol. 33, no. 2, pp. 201–228, 2007.

[10] L. Huang and D. Chiang, "Forest rescoring: Faster decoding with integrated language models," in *Proceedings of ACL*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 144–151. [Online]. Available: http://www.aclweb.org/anthology/P/P07/P07-1019

[11] D. Wu, "Stochastic inversion transduction grammars and bilingual parsing of parallel corpora," *Comput. Linguist.*, vol. 23, pp. 377–404, 1997.

[12] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of ACL*, 2003, pp. 160–167.

[13] D. Xiong, Q. Liu, and S. Lin, " "maximumentropy based phrase reordering model for statistical machine translation," in *Proceedings of COLING/ACL 2006*, 2065, pp. 521–528.

[14] A. Stolcke, "Srilm - an extensible language modeling toolkit," in *Proceedings of ICSLP*, vol. 30, 2002, pp. 901–904.

[15] D. Xiong, S. Li, Q. Liu, and S. Lin, "Parsing the penn chinese treebank with semantic knowledge," in *Proceedings of IJCNLP 2005*, 2005, pp. 70–81.

[16] E. Charniak and M. Johnson, "Coarse-to-fine-grained n-best parsing and discriminative reranking," in *Proceedings of the 43rd ACL*, 2005.

[17] L. Huang, "Forest reranking: Discriminative parsing with non-local features," in *Proceedings of ACL*, 2008.