I²R Chinese-English Translation System for IWSLT 2007

Boxing Chen, Jun Sun, Hongfei Jiang, Min Zhang, Ai Ti Aw

Department of Human Language Technology
Institute for Infocomm Research, Singapore
{bxchen, visjs, vishfj, mzhang, aaiti}@i2r.a-star.edu.sg

Abstract

In this paper, we describe the system and approach used by Institute for Infocomm Research (I²R) for the IWSLT 2007 spoken language evaluation campaign. A multi-pass approach is exploited to generate and select best translation. First, we use two decoders namely the open source Moses and an in-home syntax-based decoder to generate N-best lists. Next we spawn new translation entries through a word-based *n*-gram language model estimated on the former N-best entries. Finally, we join the N-best lists from the previous two passes, and select the best translation by rescoring them with additional feature functions.

In particular, this paper reports our effort on new translation entry generation and system combination. The performance on development and test sets are reported. The system was ranked first with respect to the BLEU measure in Chinese-to-English *open data* track.

1. Introduction

This paper describes the statistical machine translation (SMT) system and approach undertaken by the Institute for Infocomm Research (I²R) for the International Workshop on Spoken Language Translation (IWSLT) 2007. We submitted runs under the *open data* conditions for Chinese-to-English task.

A typical state-of-the-art SMT system applies two-pass search strategy [1, 2]. In the first pass, N-best translations are generated by a decoding algorithm; while in the second pass, the best translation is computed by rescoring and reranking the N-best translations with additional feature functions.

In our MT system, we introduce an intermediate (*re-generation*) pass before rescoring. The N-best lists generated separately by different decoders are consolidated first before further expanded to create a new list by applying a generative *n*-gram language model, estimated on the joined N-best lists [3]. In our experiment, three N-best lists are generated in the first pass. One is provided by an in-home syntax-based decoder. The other two are generated by Moses decoder [4] with different data preprocessing.

This paper is organized as follows. Section 2 presents the structure of our system with emphasis on the syntax-based decoder. We also describe the *n*-gram expansion, system combination and rescoring modules. Section 3 reports the experimental setups and results with discusses on the results obtained while Section 4 concludes the paper.

2. System Description

Figure 1 depicts our system architecture. A source input sentence is preprocessed with two different settings. It is then passed to the phrase-based decoder (Moses) and the syntax-based decoder to produce three N-best translation lists. All the three N-best lists go through the *n*-gram expansion to create new translation hypotheses, and formed the fourth N-best list. All four N-best lists are then combined, and the best translation is selected by rescoring approach. We will detail the following components namely phrase-based system, syntax-based system, *n*-gram expansion, system combination and rescoring.

2.1. Phrase-based system

Phrase-based statistical machine translation systems are usually modeled through a log-linear framework [5, 6] by introducing the hidden word alignment variable a [7].

$$\tilde{e}^* = \arg\max_{e,a}(\sum_{m=1}^{M} \lambda_m h_m(\tilde{e}, f, a)) \qquad (1)$$

where $\tilde{e}$ is a string of phrases in the target language, f is the source language string, $h_m(\tilde{e}, f, a)$ are feature functions, weight λ_m are typically optimized to maximize the scoring function [8].

Our two phrase-based models are based on Moses decoder with word alignment obtained from GIZA++ [9]. The translation model, lexicalized word reordering model are trained using the tools provided in the open source Moses package. Language Model is trained with SRILM toolkit [10]. It is a 5-gram LM with modified Kneser-Ney smoothing method.

The search operation is accomplished by Moses decoder. Two different N-best lists are generated, each from the same source input but preprocessed differently.

2.2. Syntax-based system

We develop a syntax-based tree-to-tree alignment model that can capture global structure distortion and global reordering [11]. The model is formally a probabilistic synchronous tree-substitution grammar (STSG) that is a collection of aligned elementary tree pairs with mapping probabilities (which are automatically learned from word-aligned bi-parsed parallel texts).

A synchronous tree-substitution grammar (STSG) $G =< \Sigma_s, \Sigma_t, N_s, N_t, S_s, S_t, P >$ is a septet, where:

- Σ_s and Σ_t are source and target terminal alphabets (POSs or lexical words), respectively, and

- N_s and N_t are source and target non-terminal alphabets (linguistic phrase tag, i.e., NP/VP…), respectively, and

- $S_s \in N_s$ and $S_t \in N_t$ are the source and target start symbols (roots of source and target parse trees), and

- P is a production rule set, where a production rule is a pair of elementary tree ($\xi_s \leftrightarrow \xi_t$) with linking relation

between leaf nodes in source elementary tree (ξs) and leaf nodes in target elementary tree (ξt). An elementary tree is a tree fragment whose leaf nodes are either non-terminal or terminal symbols.

In the rule extraction step, initial or abstract pairs of element tree (*PET*) will be extracted as the tree transfer rules.

In the decoding step, a source sentence is first parsed. All possible element trees of the source sentence are used to match their possible transfer rules. Last, based on a featured log-linear model, the optimal target tree is generated.

$$\hat{e}_1^I = \arg\max_{e_1^I, PET_1^K}(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J, PET_1^K)) \qquad (2)$$

For details of the syntax-based system, please refer to paper [11].

2.3. N-gram expansion

We utilize a so-called *re-generation* pass in our system by means of *n*-gram expansion [3]. It is applied to the N-best lists of translation hypotheses produced by different decoders but from the same given source sentence. The underlying principle behind this approach is that alternative translation hypotheses could be generated by combining the substrings occurring in the different N-best lists. An *n*-gram LM is estimated on the original N-best lists, and then new hypotheses are generated through this LM by continuously expanding the partial hypothesis. Figure 2 illustrates the idea of this approach. Once we have a partial hypothesis ($e_1 \, e_2 \, e_3 \, e_4 \, e_5$), it can be expanded by adding one word (e_6) through an *n*-gram ($e_4 \, e_5 \, e_6$) which first *n-1* words match the last *n-1* words of the hypothesis ($e_4 \, e_5$).

Figure2: Expansion of a partial hypothesis via a matching *n*-gram

Figure 3 shows a simple example of *n*-gram expansion. Suppose we have two original hypotheses as shown in Figure 3. The operation of *n*-gram expansion could generate two new hypotheses through a 3-gram LM estimated on the original hypotheses. The first new hypothesis is exactly the same as the reference which generated by combining the two substrings in italic of the original hypotheses. For the details of this approach, please refer to paper [3].

Reference	my book is in the green basket .
Orig. hyp.	*my book is in the green* case . my book is inside *the green basket* .
New hyp.	my book is in the green basket . my book is inside the green case .

Figure 3: *n*-gram expansion generates a new hypothesis which is the same as the reference.

2.4. System combination

The aim of system combination is to increase the possibility of having a better 1-best output by combining N-best lists generated by each individual system. As depicted in Figure 1, the three N-best lists generated in the first pass, together with the new hypotheses generated through *n*-gram expansion, are merged with removing duplicate hypotheses.

2.5. Rescoring model

Since hypotheses are produced from *re-generation* (*n*-gram expansion) and systems with different decoders, the local feature functions of each hypothesis are not comparable and cannot be used in rescoring. We thus exploit rich global feature functions in the rescoring models which were used in ITC-irst[1] SMT system for IWSLT 2006 [12] to compensate the loss in local feature functions. We apply the following 10 feature functions. Weights of feature functions are optimized by the tool in Moses package.

- direct and inverse IBM model 1 and 3

- competitive linking algorithm (CLA) association score, i.e. hyper-geometric distribution probabilities and mutual information [13]

- lexicalized word/block reordering probabilities [14], however, the reordering model here is trained on a CLA word-aligned training data set and word order of the hypothesis is also given by CLA as suggested in [3]

- 6-gram target LM

- 8-gram target word-class based LM, word-classes are clustered by GIZA++

- length ratio between source and target sentence

- question feature

- frequency of its n-gram (n=1,2,3,4) within the N-best translations

- n-gram posterior probabilities within the N-best translations [15]

- sentence length posterior probabilities [15]

The above rescoring feature functions could be classified into 5 groups as described below. The first seven feature functions correspond to the feature functions used in a typical phrase-based decoder, i.e. Moses.

- IBM models and CLA association score correspond to translation model

- word/block reordering probabilities is reordering model

- same target LM but with the addition of target word-class based LM

- length ratio compensates word penalty

- the last four feature functions model the translation confidence of the hypotheses

---
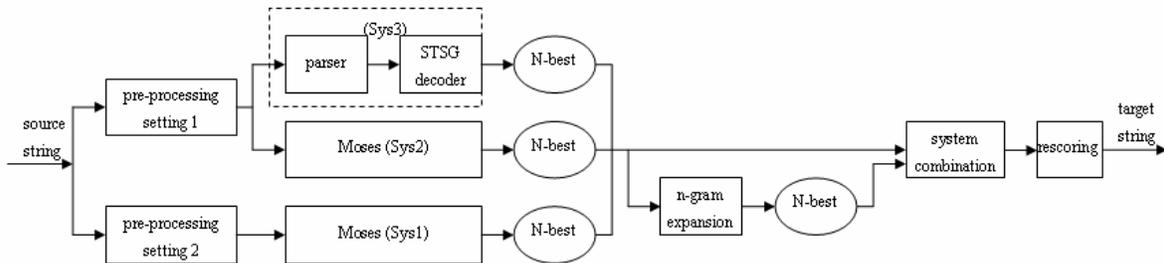
Figure 1: System structure.

Table 2: Statistics of training data.

		Sys1		Sys2		Sys3	
		ch	en	ch	en	ch	en
Parallel Training data (BTEC+HIT-corpus)	Sentences	406,122				96,832	
	Running words	4,443K	4,591K	4,537K	4,621K	922K	1,007K
	Vocabulary	69,989	61,087	53,841	62,440	24,743	21,376
Additional target data (Tanaka Corpus)	Running words	1,398K					

3. Experiments

Experiments were carried out on the *Basic Traveling Expression Corpus* (BTEC) Chinese-English data [16] augmented with *HIT-corpus[1]*. BTEC is a multilingual speech corpus which contains sentences coming from phrase books for tourists. 40K sentence-pairs are supplied for IWSLT 2007. *HIT-corpus* has 500K sentence-pairs in total. We selected 360K sentence-pairs that are more similar to BTEC data. For efficiency purpose, the syntax-based system used a smaller training data set which contains supplied BTEC data and the *Olympic data[2]* in *HIT-corpus*. Additionally, the English sentences of *Tanaka* corpus[3] were also used to train our language model.

3.1. Preprocessing

Preprocessing includes Chinese word segmentation, tokenization, parsing and transformation of numbers from textual-form to digit-form (txt-to-digit) and lower-casing.

We used the following tools in the preprocessing: (1) Stanford parser[4] for both Chinese and English, (2) ICTCLAS[5] word segmentation tool developed in ICT [17] and DP-based word segmentation script[6] with LDC Chinese words list (LDC-SEG).

As mentioned in section 2.1, we set up two systems based on Moses with different preprocessing settings. Their different settings are showed in Table 1 and named "Sys1" and "Sys2" respectively. We refer the syntax-based system as "Sys3". In particular, "Sys1" used LDC-SEG word segmentation while "Sys2" and "Sys3" used ICTCLAS. Parsing was only performed on "Sys3". "Sys1" performed

txt-to-digit operation, while not the other two. The reason why we do not use txt-to-digit conversion in "Sys2/3" is due to the following two considerations: 1) to generate more distinct N-best lists from the two Moses based systems with the help of different representations of Chinese number; 2) to take both advantages of Chinese number characters in textual-form and digit-form since given context, some Chinese char can only be written in text form and some are correct in both form. For example, Chinese word *wan4yi1* (means "in case") can only be written in text form although it is composed of two Chinese number characters *wan4* (ten thousands) and *yi1* (one).

Detailed statistics of the preprocessed training, development and test data are shown in Table 2 and 3.

Table 1: Preprocessing operations applied; "x" means that operation is performed; "L" means LDC segmentation tool; "I" means ICTCLAS.

Preprocessing	Sys1		Sys2		Sys3	
	ch	en	ch	en	ch	en
Tokenization	L	x	I	x	I	x
Parsing	-	-	-	-	x	x
Txt-to-digit	x	x	-	-	-	-
Lower-casing	-	x	-	x	-	x

3.2. Postprocessing

The evaluation of IWSLT'07 is case sensitive. To reduce data sparseness, we lowercase the target language in the preprocessing step. Thus, a case restoration post-processing step is required to reinstate the correct case information.

We followed the instruction[7] provided by IWSLT'06 organizers to do case restoration. The module recovers word case information for proper names and the beginning word of a sentence. The model was trained on the same data which we used to train the language model.

---

[1] http://mitlab.hit.edu.cn/

[2] This part is the same as another corpus released by ChineseLDC: http://www.chineseldc.org/EN/purchasing.htm (code: 2004-863-008)

[3] http://www.csse.monash.edu.au/~jwb/tanakacorpus.html

[4] http://www-nlp.stanford.edu/

[5] http://www.nlp.org.cn/project/project.php?proj_id=6

[6] http://projects.ldc.upenn.edu/Chinese/LDC_ch.htm

---

[7] http://www.slc.atr.jp/IWSLT2006/

Case restoration was done after the rescoring pass using *disambig* tool from SRILM toolkit.

Table 3: Statistics of development and testing data.

		Chinese		English
		Sys1	Sys2/3	
Dev1 (CSTAR'03)	Sentences	506		506×16
	words	3,402	3,469	65,615
Dev2 (IWSLT'04)	Sentences	500		500×16
	Words	3,502	3,631	64,884
Dev3 (IWSLT'05)	Sentences	506		506×16
	Words	3,772	3,985	55,935
Dev4 (DEV'06)	Sentences	489		489×7
	Words	5,896	5,976	45,449
Dev5 (TEST'06)	Sentences	500		500×7
	Words	6,296	6,384	51,227
Test'07	Sentences	489		489×6
	Words	3,190	3,394	22,574

3.3. Experiment Settings

For "Sys1" and "Sys2", we extracted 1,000-best translations for each source input, with duplicates found in each N-best list. Here, we did not use the Moses option "distinct" to generate distinct N-best hypotheses. This is because 1) generating distinct hypotheses are very time-consuming; 2) since input sentences are relatively short, distinct 1,000-best may contain "bad" hypotheses which may hurt the LM which will be used in our n-gram expansion step; 3) generally, duplicated hypotheses imply higher translation confidence, which could improve the generative LM used in our n-gram expansion step. We extracted 500-best hypotheses from "Sys3" with no duplicates. We then selected 1,500-best entries from *re-generation*, also with no duplicate. All of these N-best entries are then combined and with duplicates removed.

We carried out two series of experiments. The first is on using CSTAR'03 (dev1) set as development set, and IWSLT'04 (dev2) and IWSLT'05 (dev3) as test sets. The second is on using DEV'06 (dev4) clean text set as development set and TEST'06 (dev5) clean text as test set. The original source sentences of dev4 and dev5 do not contain punctuations. We did punctuation insertion before feeding them to the decoders. Following the instructions provided by IWSLT'06 organizers again, the punctuation insertion was performed using *hidden-ngram* command in SRILM toolkit. The development sets dev4/5 with punctuation insertion were added to the training data in the first experiment while dev1-3 were added to the training set in the second experiments.

Tables 4 and 5 show the results of the two experiments. All the scores are computed based on case insensitive and with punctuation. In Table 4 and 5, the rows "Sys1/2/3" indicate the baseline performance of the three systems; "Comb" refers to the results of the final translation output with *re-generation*, system combination and rescoring incorporated.

For comparison purpose, we also evaluated rescoring on "Sys1" N-best list and combination of "Sys1" and "Sys2" N-best lists. We refer them as "Resc1" and "Resc2"

respectively. Note that on top of the same global feature functions used in "Comb" as mentioned in section 2.5, the local feature functions used during decoding are also involved in rescoring "Resc1/2".

Table 4: BLEU% and NIST scores of the experiment one; with punctuation, no case.

	Dev set (Dev1)		Test Set 1 (Dev2)		Test Set 2 (Dev3)	
	BLEU	NIST	BLEU	NIST	BLEU	NIST
Sys1	56.03	8.733	56.31	9.337	62.06	9.959
Sys2	55.40	8.543	56.03	8.954	62.34	9.687
Sys3	45.61	8.138	51.21	8.988	54.43	9.407
Resc1	57.60	9.017	56.98	9.527	64.09	10.376
Resc2	58.40	8.959	59.87	9.510	64.16	9.991
Comb	57.51	9.036	59.76	9.743	64.69	10.391

Table 5: BLEU% and NIST scores of the experiment two; with punctuation, no case.

	Dev Set (Dev4)		Test Set (Dev5)	
	BLEU	NIST	BLEU	NIST
Sys1	29.98	7.468	29.10	7.103
Sys2	28.50	7.223	27.67	6.789
Sys3	24.10	6.583	23.50	6.362
Resc1	31.40	7.637	30.22	7.155
Resc2	32.02	7.528	30.90	7.124
Comb	34.03	7.732	32.54	7.276

3.4. Performance discussion

Since "Sys1" and "Sys2" used the same training data and decoder, their performances are comparable. The performance of "Sys2" is a little worse than "Sys1" in experiment two, probably due to the reason that much more numbers occur in dev4/5 than that in dev1/2/3. "Sys3" shows lower performance than Moses. This may be due to two reasons: 1) less training date used in "Sys3" (see Table 2), and 2) more errors in parsing spoken language since the parser we used is trained on LDC Chinese/English Treebank (which is mostly collected from newspaper domain).

Rescoring the N-best list of "Sys1" (Resc1) has obtained relative performance improvement of 1.2% (dev2) and 4.8% (dev4) in BLEU score. "Resc2" has achieved even better performance improvement in BLEU score but lost in NIST score when compared with "Resc1". The reason might be that we tuned the weights according to BLEU score and "Sys2" has a lower NIST scores on all development and test sets.

It is hard to compare the performance of "Comb" with other systems, and choose the absolutely "best" system. "Comb" achieved the best NIST score for all the five development and test sets, but not for BLEU score on dev1 and dev2, especially on dev1. Another interesting phenomena is that in experiment one, the performance of "Comb" and "Resc1/2" are very similar, especially in comparing the value of $100 \times BLEU+4 \times NIST$ [18]. However, "Comb" is significantly better than others in experiment two (Table 5). By evaluating against "Resc2", relative improvement of 6.3%

on dev4 and 5.3% on dev5 of BLEU score have been obtained in "Comb".

To better interpret this phenomena, let us take a look into the number of distinct N-best entries of each system in Table 6. We can see that *re-generation* enlarged the N-best lists much more significantly on dev4 and dev5 than on dev1, dev2 and dev3. It means that for dev1-3 sets, most of the "good" hypotheses generated during *n*-gram expansion already existed in the N-best lists of Sys1-3, because the input sentences are short (average about 7 words). But if the input sentences are longer (average about 12 words in dev4 and dev5), much more "good" hypotheses could be generated during *n*-gram expansion. This could also be proven by observing Table 7 which shows the number of new generated hypotheses in the final translation output. Dev4 and dev5 selected much more new generated hypotheses in the final translation output. Since in the rescoring stage, "Comb" only used the global feature functions, but "Resc1/2" also involved the local feature functions used in decoding. It seems that the benefit of new hypotheses can only compensate the loss in local features for the short sentences in experiment one, but for the longer sentences in experiment two, the benefit of new hypotheses exceeded the loss of local feature functions.

Table 6: Number of N-best entries.

	Dev1	Dev2	Dev3	Dev4	Dev5
Sys1	271K	249K	223K	188K	187K
Sys1+2	450K	380K	335K	369K	377K
Sys1+2+3	675K	+596K	554K	612K	629K
Comb	882K	777K	722K	1025K	1056K

Table 7: Number of new generated hypotheses in the final translation output.

	Dev1	Dev2	Dev3	Dev4	Dev5
#new hypo	29	18	12	59	74

From another viewpoint, phrase-based system cannot handle global word reordering very well, since the distortion is limited. But *n*-gram expansion permits long distance word movements through a low-order LM (e.g. a bigram LM). So obviously, *re-generation* could benefit longer source sentences more than shorter sentences. Table 8 shows the average source sentence length and relative improvements of system "Comb" against system "Resc2". The relative improvement increases with the average source sentence length almost consistently.

Table 8: Source sentence average length and relative improvements on BLEU score (%).

	Dev1	Dev2	Dev3	Dev4	Dev5
Length	6.7	7.0	7.5	12.1	12.6
Δ	-1.5	-0.2	0.8	6.3	5.3

3.5. Official scores on test set

Table 9 shows the official scores on the test set as reported by the IWSLT'07 organizers. Only BLEU scores were reported. We submitted two runs performed by "Resc2" (primary run)

and "Comb" (second run). During submission, we observed that the test set (classic task) is more similar to dev1/2/3 (classic task) as compared to dev4/5 (challenge task), and "Resc2" also produces better BLEU score on dev1 and dev2 than "Comb". We also noticed that the average source sentence length of Test'07 is close to that of dev1 (6.5 vs. 6.7) than all other development sets, so we expect similar performance behavior for Test'07 and thus submit "Resc2" as our primary run. Note that dev1 (CSTAR'03) was used as development set for both runs.

Table 9: Official scores of test set (case sensitive).

	BLEU%
Run1	40.77
Run2	39.42

3.6. Performance on test set using official data only

After official submission, we have also experimented on test set by using the official data only. Dev1 was used as the development set, and the same as before dev2-5 were added to the training data. Table 10 reports the performance when using official data only, which further demonstrates the effectiveness of our proposed methods (system combination and rescoring).

Table 10: Scores of test set on official data (case sensitive).

	BLEU%	NIST
Resc2	38.67	6.740
Comb	37.03	6.756

4. Conclusions

This paper described the I²R SMT system that was used in the IWSLT 2007 evaluation campaign. We use a multi-pass approach. N-best lists of translations are generated in the first pass; then the N-best lists are enlarged by means of *n*-gram expansion; finally, rescoring and re-ranking are applied to select best translation.

5. Acknowledgements

6. References

[1] P. Koehn, F. J. Och and D. Marcu, "Statistical Machine Phrase-based Translation". *Proceedings of HLT/NAACL*, pp 127-133, Edmonton, Canada, 2003.

[2] M. Federico and N. Bertodi, "A word-to-phrase statistical translation model", *ACM Transaction on Speech Language Processing*, vol. 2, no. 2, pp. 1-24, 2005.

[3] B. Chen, M. Federico and M. Cettolo, "Better N-best Translation through Generative *n*-gram Language Model", *Proceeding of MT Summit XI*, Copenhagen, Denmark, September, 2007.

[4] 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", *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Poster Session, pp. 177-180, Prague, Czech Republic, June 2007.

[5] A. Berger, S. Della Pietra and V. Della Pietra, "A Maximum Entropy Approach to Natural Language Processing", *Computational Linguistics*, vol. 22, no. 1, March 1996.

[6] F. J. Och, H. Ney, "Discriminative training and maximum entropy models for statistical machine translation", *Proceeding of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp 295-302, PA Philadelphia, USA, July, 2002.

[7] P. F. Brown, V. J. Della Pietra, S. A. Della Pietra & R. L. Mercer, "The Mathematics of Statistical Machine Translation: Parameter Estimation", *Computational Linguistics*, 19(2) 263-312, 1993.

[8] F. J. Och, Minimum error rate training in statistical machine translation. "*Proceeding of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*", pages 160--167, Sapporo, Japan, July 2003.

[9] F. J. Och, and H. Ney, "A Systematic Comparison of Various Statistical Alignment Models", *Computational Linguistics*, volume 29, number 1, pp. 19-51 March 2003.

[10] A. Stolcke, "SRILM -- an extensible language modeling toolkit", *Proceeding of International Conference on Spoken Language Processing*, 2002.

[11] M. Zhang, H. Jiang, A. T. Aw, J. Sun, S. Li, C. L. Tan, "A Tree-to-Tree Alignment-based Model for Statistical Machine Translation", *Proceeding of MT Summit XI*, Copenhagen, Denmark, September, 2007.

[12] B. Chen, R. Cattoni, N. Bertoldi, M. Cettolo and M. Federico, "The ITC-irst SMT System for IWSLT-2006", *Proceeding of International Workshop on Spoken Language Translation*, Kyoto, Japan, November, 2006.

[13] B. Chen, M. Federico. "Improving Phrase-Based Statistical Translation through Combination of Word Alignment", *Proceedings of FinTAL - 5th International Conference on Natural Language Processing*, Springer Verlag, LNCS, Turku, Finland. August 23-25, 2006.

[14] B. Chen, M. Cettolo and M. Federico, "Reordering Rules for Phrase-based Statistical Machine Translation", *Proceeding of International Workshop on Spoken Language Translation*, pp. 182-189, Kyoto, Japan, November, 2006.

[15] R. Zens and H. Ney, "N-gram Posterior Probabilities for Statistical Machine Translation", *Proceedings of the HLT-NAACL Workshop on Statistical Machine Translation*, pp. 72-77, New York City, NY, June 2006.

[16] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, "Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world", *Proceeding of LREC-2002: Third International Conference on Language Resources and Evaluation*, pp.147-152, Las Palmas de Gran Canaria, Spain, 27 May - 2 June 2002.

[17] H. Zhang, H. Yu, D. Xiong and Q. Liu, "HHMM-based Chinese Lexical Analyzer ICTCLAS", *Proceedings of SigHan2003 Workshop*, pp.184-187, Sapporo, Jappan, 2003.

[18] B. Chen, R. Cattoni, N. Bertoldi, M. Cettolo and M. Federico, "The ITC-irst SMT System for IWSLT-2005", *Proceeding of International Workshop on Spoken Language Translation*, pp.98-104, Pittsburgh, USA, October, 2005.