I^{2}R Multi-Pass Machine Translation System for IWSLT 2008

Boxing Chen, Deyi Xiong, Min Zhang, Aiti Aw, Haizhou Li

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

Abstract

In this paper, we describe the system and approach used by the Institute for Infocomm Research (I^{2}R) for the IWSLT 2008 spoken language translation evaluation campaign. In the system, we integrate various decoding algorithms into a multi-pass translation framework. The multi-pass approach enables us to utilize various decoding algorithm and to explore much more hypotheses. This paper reports our design philosophy, overall architecture, each individual system and various system combination methods that we have explored. The performance on development and test sets are reported in detail in the paper. The system has shown competitive performance with respect to the BLEU and METEOR measures in Chinese-English Challenge and BTEC tasks.

1. Introduction

This paper describes the machine translation (MT) system and approach explored by the Institute for Infocomm Research (I^{2}R) for the International Workshop on Spoken Language Translation (IWSLT) 2008. We submitted runs under the *open data* conditions for Chinese-to-English BTEC and Challenge tasks.

System combination [1, 2, 3] has demonstrated its advantage in the recent machine translation evaluation campaign [4, 5]. In our system, a multi-pass SMT approach is exploited which consists of decoding, regeneration, rescoring and system combination. First, multiple systems based on different translation strategies are used to generate various N-best lists. This aims to leverage on the strength of different translation methods. Then three kinds of different system combination methods are applied in a two-stage procedure to find the 1-best translation.

Figure 1 depicts our system architecture. First, we use three decoders, namely Moses [6] (an open source phrase-based MT system), JosHUa [7] (a hierarchical phrase-based translation system) and Tranyu [8] (an in-house linguistically-annotated BTG-based decoder) to generate 2N-best lists of hypotheses for each decoder. Then each 2N-best lists are rescored and re-ranked with additional feature functions. The 1-best and top N-best lists of these re-ranked lists are then used in system combination[1].

Secondly, we construct system combination in two-stage. In the first stage, two strategies are applied. The first strategy is n-gram expansion by which we spawn new translation entries through a word-based n-gram language model estimated on the input hypotheses. Then input hypotheses and the newly-generated hypotheses by n-gram expansion are simply combined and rescored with additional feature functions. The second strategy is simply to cascade two N-best lists, where both of which are generated by Moses but with different input data preprocessing. Finally, in the second stage of system combination, a simple weighted voting algorithm is adopted to re-rank all the previously generated 1-best.

The rest of the paper is organized as follows. Section 2 presents each individual SMT model used in our system. Section 3 details the rescoring models. Section 4 describes three system combination strategies: n-gram expansion, simple cascading and weighted voting. Section 5 reports the experimental setups and results while Section 6 concludes the paper.

2. The SMT Models

To integrate the advantages of the state-of-the-art translation methods, we use three different SMT models, phrase-based, hierarchical phrase-based and linguistically-annotated BTG-based in the first pass to generate N-best hypotheses. The three methods share the some common features: word alignment of training data obtained from GIZA++ [9], Language model(s) (LM) trained using SRILM toolkit [10] with modified Kneser-Ney smoothing method [11].

2.1. Phrasal translation system

Phrase-based SMT systems are usually modeled through a log-linear framework [12]. By introducing the hidden word alignment variable a [13], the optimal translation can be searched for based on the following criterion:

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

where $\tilde{e}$ is a string of phrases in the target language, $\tilde{f}$ is the source language string of phrases, $h_m(\tilde{e}, \tilde{f}, a)$ are feature functions, weights λ_m are typically optimized to maximize the scoring function [14].

Our phrasal translation system is based on the Moses open source package [6]. IBM word reordering constraints [15] are applied during decoding to reduce the computational

[1] Since our system combination method n-gram expansion [3] is based on a generative language model that is trained on the input hypotheses lists, the hypotheses quality is very important to the performance of the n-gram expansion method. Therefore, we filter out N-worse hypotheses from the 2N-best lists before passing them to the n-gram expansion model.

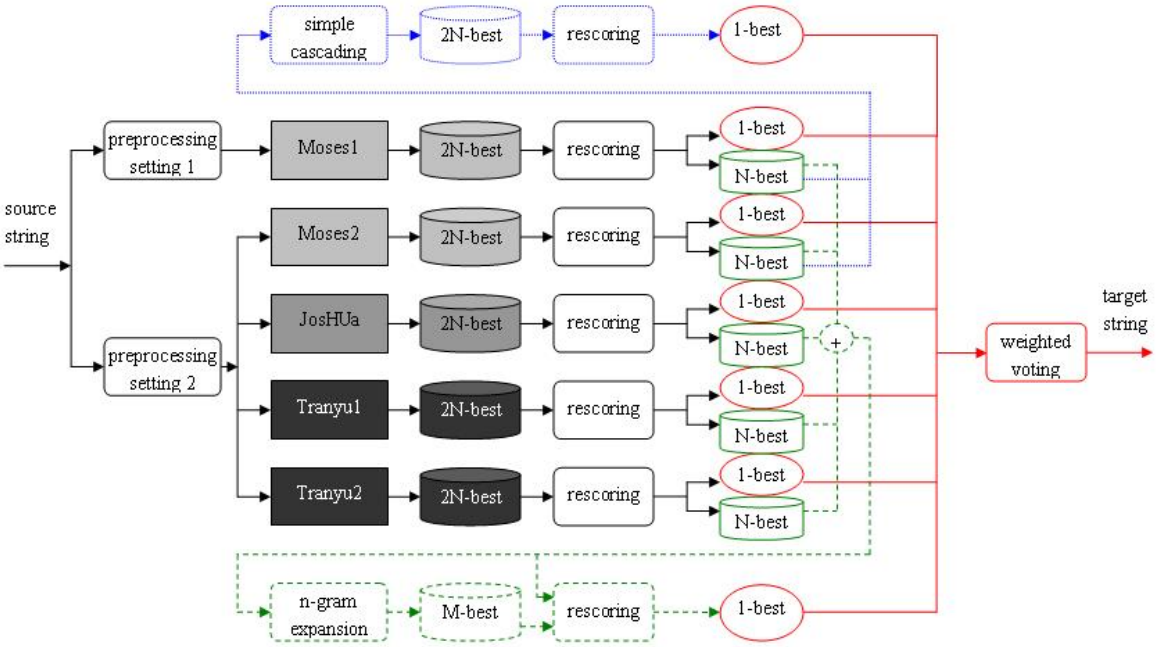

Figure 1: system architecture

complexity. The other models and feature functions employed by Moses decoder are:

- Translation model(s) (TM), direct and inverse phrase/word based translation model
- Distortion model, which assigns a cost linear to the reordering distance, the cost is based on the number of source words which are skipped when translating a new source phrase
- Lexicalized word reordering model [16] (RM)
- Word and phrase penalties, which count the numbers of words and phrases in the target string

The translation model, reordering model and feature weights are trained and optimized using Moses training and tuning toolkits. Two different N-best lists are generated by the same Moses decoder with the same source input but different preprocessing.

2.2. Hierarchical phrase-based translation system

Hierarchical phrase-based translation method is a typical formally syntax-based translation modeling method. Empirically, it has demonstrated better performance than the phrase-based method because it permits phrases with gaps by generalizing the normal phrase-based models [17, 7]. Formally, the hierarchical phrase-based translation model is a weighted synchronous context free grammar. In our system combination framework, for the hierarchical phrase-based translation component, we use the default setting as discussed in [17] for training and tuning and use JosHUa [7]'s implementation for decoding.

2.3. Linguistically annotated BTG-based system

Tranyu is an in-house formally and linguistically syntax-based SMT system, which adopts the bracketing transduction grammars (BTG) as the fundamental framework for phrase translation and reordering. The BTG lexical rules (A --> x/y) are used to translate source phrase x into target phrase y while the BTG merging rules (A --> [A, A]|<A, A>) are used to combine two neighboring phrases with a straight or inverted order. All these rules are weighted with various features, such as most of phrase translation probabilities used in phrase-based system and reordering features, in a log-linear form. We incorporate two individual maximum entropy based reordering models into the log-linear translation model to predict phrase orders. The first model uses boundary words of neighboring phrases as features [8], which we call the boundary words based reordering model (BWR). The second model uses linguistic annotations of each BTG node as features, which are automatically learned by projecting source-side parse trees onto the corresponding binary trees generated by BTG. We call the second model the linguistically annotated reordering model (LAR). Based on these two reordering models, we developed two variations of Tranyu. The first variation Tranyu1 only uses the BWR model [8] while the second variation Tranyu2 uses both BWR and LAR models [18].

3. Rescoring models

Rescoring operation plays a very important role in our system. A rich global feature functions set benefits our system greatly. The rescoring models are the same ones which were used in our SMT system for IWSLT 2007 [4]. We apply the following feature functions. Weights of feature functions are optimized by the MERT tool in Moses package.

- direct and inverse IBM model 1 and 3
- association scores, i.e. hyper-geometric distribution probabilities and mutual information
- lexicalized reordering rule [19]
- 6-gram target language model and 8-gram target word-class based LM, word-classes are clustered by GIZA++
- length ratio between source and target sentence
- question feature [20]
- Linear sum of n-grams (n=1,2,3,4) relative frequencies within all translations [20]
- n-gram and sentence length posterior probabilities within the N-best translations [21]

4. System combination

In our system, three different system combination strategies are used in a two-stage procedure to find the final translation. They are simple cascading, n-gram expansion in the first stage, and weighted voting in the second stage.

4.1. N-gram expansion

N-gram expansion [3] combines the sub-strings occurred in the original N-best translations to generate new hypotheses. Firstly, all n-grams from the original N-best translations are collected. Then the partial hypotheses are continuously expanded by appending a word through the n-grams collected in the first step.

During the new hypotheses generation step, the translation outputs are computed through a beam-search algorithm with a log-linear combination of the feature functions. In addition to n-gram frequency and n-gram posterior probability used in [3], we follow the suggestion of [22] and also use language model, direct/inverse IBM model 1, and word penalty in this work.

4.2. Simple cascading

Local feature functions used by different individual system are not comparable, and thus cannot be used in rescoring. To take the advantage of the rich and powerful local feature functions, we conduct alternative system combination method, which we called simple cascading. We just simply combine the outputs of Moses1 and Moses2, and then rescore the combined list based on local feature functions and global rescoring models as discussed in section 3.

4.3. Weighted voting

As shown in Figure 1, given all 1-best hypotheses generated from different systems, the final 1-best translation is selected by weighted voting. In our weighted voting, a binary feature function is used to indicate the system in which hypothesis is generated from. Note that we have five individual systems and two combined systems. The feature weight of each system is tuned over the development set. If all the weights are set to 1, then we call it simple voting.

5. Experiments

We participated Chinese-to-English BTEC task (BT) and Challenge task (CT) in *open data* track for IWSLT 2008.

5.1. Preprocessing

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

We used two tools for word segmentation: (1) ICTCLAS[1] developed in ICT [23] and DP-based word segmentation script[2] with LDC Chinese words list (LDC-SEG).

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

preprocessing	Setting 1		Setting 2	
	ch	en	ch	en
Tokenization	L	x	I	x
Txt-to-digit	x	x	-	-
Lower-casing	-	x	-	x

As shown in Figure 1, we set up two systems based on Moses with different preprocessing settings. Their different settings are showed in Table 1 and named “Setting 1” and “Setting 2” respectively by following our IWSLT-2007 system [4]. In particular, Setting1 used LDC-SEG word segmentation while Setting 2 used ICTCLAS. Setting 1 performed txt-to-digit operation, while Setting 2 does not.

5.2. Postprocessing

The evaluation of IWSLT’08 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 recover the correct case information.

We followed the instruction[3] provided by IWSLT’08 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.

Case restoration was done on the final MT output using *disambig* tool from SRILM toolkit.

5.3. Data

Experiments were carried out on the *Basic Traveling Expression Corpus* (BTEC) Chinese-English data [24]

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

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

[3] http://www.slc.atr.jp/IWSLT2008/

augmented with *HIT-corpus*[1] and *Olympic-corpus*[2], *PKU-corpus*[3] from Chinese LDC. BTEC is a multilingual speech corpus which contains sentences coming from phrase books for tourists. 20K sentence-pairs are supplied for IWSLT 2008. *HIT-corpus* has 132K sentence-pairs in total, and is mainly multi-source Chinese-English parallel corpus; *Olympic-corpus* contains 54K sentence-pairs mainly in sport domain and travelling domain; *PKU-corpus* has about 200K sentence-pairs, and is a domain-balanced corpus. Additionally, the English sentences of *Tanaka-corpus*[4] were also used to train our language model. We just simply joined all the data together due to a big change of *HIT-corpus* in the last minute.

Moreover, there are 7 development sets provided for challenge task and 6 development sets for BTEC task. The first development set *IWSLT08_BTEC.devset1_CSTAR03* (CSTAR) was used as the tuning set in BTEC task, while development set *IWSLT08_CT_CE.devset* (DEV08) was used as the tuning set in challenge task. We also add these development sets to the training data for the official testing. During the tuning stage, two training sets are set up for two tasks. Thus, we name four training sets, they are: training set for BTEC task developing (BT-Dev-Train), training set for BTEC task testing (BT-Tst-Train), training set for challenge task developing (CT-Dev-Train), training set for challenge task testing (CT-Tst-Train). Detailed statistics of all training data with preprocessing Setting 2 are shown in Table 2, and the difference of the above four training sets is shown in Table 3.

The test sets of both tasks have two types of input: transcription of automatic speech recognition (ASR), and correct recognition result (CRR). For the ASR input, we used 1-best recognition results. The original source sentences of these test sets do not contain punctuations. We did punctuation insertion before feeding them to the decoder. Following the instructions provided by IWSLT'08 organizers, the punctuation insertion was performed using *hidden-ngram* command in SRILM toolkit.

Detailed statistics of the development and test data with preprocessing setting 2 are shown in Table 4.

Table 2: Statistics of data used in training.

data		Chinese	English
Supplied data (BTEC)	Sent	19,972	
	Words	172K	182K
	Vocab.	8,415	8,361
Additional data (all 3 corpora)	Sent.	379,065	
	Words	4,834K	5,036K
	Vocab.	57,055	75,156
AllDev data (all 7 sets)	Sent.	6,472	
	Words	56K	61K
	Vocab.	3,241	3,669
Tanaka corpus	Words	-	1,398K

Table 3: Difference between four training sets

Training set	Description
BT-Dev-Train	Supplied + Additional + AllDev – DEV08 – CSTAR
BT-Tst-Train	BT-Dev-Train + CSTAR
CT-Dev-Train	Supplied + Additional + AllDev – DEV08
CT-Tst-Train	CT-Dev-Train + DEV08

Table 4: Statistics of development and testing data.

task	type			Chinese	English
BT	Dev (CSTAR)		Sent.	506	506×16
			words	3,552	65,518
	Test		Sent.	507	-
		ASR	Words	3,567	-
		CRR	Words	3,534	-
CT	Dev (DEV08)		Sent.	246	246×7
			Words	1,617	14,295
	Test		Sent.	504	-
		ASR	Words	3,128	-
		CRR	Words	3,079	-

5.4. Results

Our evaluation metrics are BLEU [25] and NIST score, which are to perform n-grams matching up to $n = 4$. Please note that all the scores on dev sets are computed on case insensitive text including punctuation.

5.4.1. *Effect of additional data*

Table 5 shows the performances on development sets by incrementally adding the dev data and additional data to the official supplied data. It suggests both dev and additional data have improved the BLEU score significantly on both tasks.

Table 5: BLEU% , NIST scores for different training data set, with punctuation, no case.

Tasks	CSTAR		DEV08	
Score	BLEU	NIST	BLEU	NIST
supplied data	40.96	7.23	36.12	6.13
+dev data	45.76	7.62	42.29	6.59
+additional data	50.98	8.43	44.92	6.02
all data	52.28	9.04	46.45	6.46

5.4.2. *Baseline and rescoring*

For the five individual system, we extracted 5,000-best translations for each source input, without removing duplications and then a rescoring pass is applied to choose 2,500-best and 1-best for each system. To have a better N-best hypotheses list for training the LM which will be used in the next n-gram expansion step. Here, 1) we did not use the Moses option "distinct" to generate distinct N-best hypotheses because, generally speaking, duplicated

[1] http://mitlab.hit.edu.cn/index.php/resources

[2] http://www.chineseldc.org/EN/index.htm 2004-863-008

[3] http://www.chineseldc.org/EN/index.htm CLDC-LAC-2003-006

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

hypotheses imply higher translation confidence, which could improve the generative LM; 2) top 5,000-best translations may contain “very bad” hypotheses, thus we filter out 2,500-worse hypotheses from the original 5,000-best hypotheses. The size of N-best hypotheses list (2,500 in this work) is determined by observing the n-gram expansion performance on dev set.

Tables 6 shows the results of all systems’ baseline and rescoring output on two development sets.

Table 6: BLEU% and NIST scores of baseline and rescoring systems; with punctuation, no case.

Tasks		CSTAR		DEV08	
Score		BLEU	NIST	BLEU	NIST
Base	Moses1	52.20	8.69	46.19	6.52
	Moses2	52.28	**8.84**	46.45	6.56
	JosHUa	**52.62**	8.63	**47.77**	6.33
	Tranyu1	51.94	8.77	46.87	6.66
	Tranyu2	52.21	8.69	47.55	**6.99**
Resc	Moses1	54.08	8.93	49.28	6.76
	Moses2	**54.29**	**8.99**	49.45	**6.94**
	JosHUa	**54.29**	8.76	49.53	6.34
	Tranyu1	54.07	8.83	49.24	6.69
	Tranyu2	54.10	8.82	**49.76**	6.89

Comparing the performance of all baselines, hierarchical phrase-based system JosHUa achieved best BLEU score on both tasks, however, Moses2 got best NIST score on CSTAR set, and Tranyu2 obtained best NIST score on DEV08 set.

After rescoring, all BLEU scores have been improved. Moses2 and JosHUa achieved best BLEU score on CSTAR, and Tranyu2 obtained best BLEU on DEV08 set. Moses2 again got best NIST score on both CSTAR and DEV08 sets.

5.4.3. System combination

We cascaded two 2,500-best hypotheses lists from each Moses-based system for rescoring which produced about 0.9 BLEU score on CSTAR set (from 54.29 of rescoring Moses2 to 55.16) and more than 1.5 BLEU score on DEV08 set (from 49.45 of rescoring Moses2 to 50.97).

N-gram expansion achieved similar improvement over the best single system and got higher BLEU than that of simple cascading. In our IWSLT07 system [4], simple cascading outperformed n-gram expansion on dev sets whose sentences are short, such as CSTAR. In this year, the improvement of n-gram expansion may be due to the reasons as mentioned in section 4.1, we have applied more feature functions during the step of new hypotheses generation in this work. Over the best single system, n-gram expansion obtained about 1.3 BLEU score on CSTAR (from 54.29 of rescoring Moses2 to 55.61), 2.2 BLEU score on DEV08 set (from 49.76 of rescoring Tranyu2 to 52.03).

Simple voting further improved the performance, 0.8 BLEU for CSTAR, 0.6 BLEU for DEV08. Weighted voting achieved best performance on BLEU score for both dev sets.

Table7: BLEU% and NIST scores of system combination; with punctuation, no case.

System combination	CSTAR		DEV08	
	BLEU	NIST	BLEU	NIST
simple cascading	55.16	8.92	50.97	**6.88**
n-gram expansion	55.61	8.95	52.03	6.86
simple voting	56.40	8.92	52.66	6.86
weighted voting	**56.62**	**8.96**	**53.33**	6.83

5.4.4. Official scores on test set

Table 8 shows the official scores on the test set as reported by the IWSLT’08 organizers.

Table 8: Official scores (BLEU%, NIST and METEOR) of test sets; with punctuation , case sensitive.

run	task		BLEU	NIST	METEOR
Primary (weighted voting)	BT	ASR	**43.57**	6.87	**0.6017**
		CRR	**49.26**	7.65	**0.6446**
	CT	ASR	39.38	5.96	0.6142
		CRR	46.89	**6.66**	**0.6560**
Contrast1 (simple voting)	BT	ASR	42.45	6.81	0.5953
		CRR	48.12	7.56	0.6372
	CT	ASR	39.12	5.90	0.6087
		CRR	**47.87**	6.65	0.6558
Contrast2 (simple cascading)	BT	ASR	42.93	**6.94**	0.5927
		CRR	48.64	**7.75**	0.6348
	CT	ASR	37.38	5.79	0.5967
		CRR	45.67	6.55	0.6399
Contrast3 (n-gram expansion)	BT	ASR	42.16	6.47	0.5918
		CRR	47.30	7.31	0.6357
	CT	ASR	**39.54**	**6.05**	**0.6157**
		CRR	46.04	6.63	0.6546

6. Conclusions

This paper described I^{2}R’s SMT system that was used in the IWSLT 2008 evaluation campaign. We use a multi-pass approach. N-best lists of translations are generated in the first pass; then two system combination methods: simple cascading and n-gram expansion are applied; finally, weighted voting is used to select best translation.

7. References

[1] E. Matusov, N. Ueffing, and H. Ney. 2006. “Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment”. In *Proceeding of EACL-2006*, Trento, Italy.

[2] A. Rosti, N. F. Ayan, B. Xiang, S. Matsoukas, R. Schwartz and B. Dorr. “Combining Outputs from Multiple Machine Translation Systems”. In *Proceeding of NAACL-HLT-2007*, pp. 228-235. Rochester, NY.

[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] B. Chen, J. Sun, H. Jiang, M. Zhang and A. Aw. "I2R Chinese-English Translation System for IWSLT-2007", *Proceeding of IWSLT 2007*. pp. 55-60. Oct. Trento, Italy.

[5] The MSR-NRC-SRI MT System for NIST Open Machine Translation 2008 Evaluation. Available at http://research.microsoft.com/~xiaohe/publication/NIST_MT08_sys_desc_MSR-NRC-SRI_Chinese.pdf

[6] 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 *Proceedings of ACL-2007*. pp. 177-180, Prague, Czech Republic. 2007.

[7] Z. Li and S. Khudanpur. "A Scalable Decoder for Parsing-based Machine Translation with Equivalent Language Model State Maintenance." In *Proceedings of ACL SSST 2008*.

[8] D. Xiong, Q. Liu, and S. Lin. "Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation". In *Proceedings of COLING-ACL 2006*, Sydney, Australia.

[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] S. F. Chen and J. T. Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. *Technical Report TR-10-98*, Computer Science Group, Harvard University.

[12] F. J. Och, and H. Ney. "Discriminative Training and Maximum Entropy Models for Statistical Machine Translation." In *Proceeding of ACL-2002*. 2002.

[13] 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.

[14] F. J. Och. "Minimum error rate training in statistical machine translation." In *Proceedings of ACL-2003*. Sapporo, Japan. 2003.

[15] A. L. Berger, P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, A. S. Kehler, and R. L. Mercer. "Language translation apparatus and methods using context-based translation models". US Patent 5,510,981. 1996.

[16] P. Koehn, A. Axelrod, A. B. Mayne, C. Callison-Burch, M. Osborne and D. Talbot. "Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation." In *Proceeding of IWSLT-2005*.

[17] D. Chiang. "Hierarchical phrase-based translation." *Computational Linguistics*, 33(2):201-228. 2007.

[18] D. Xiong, M. Zhang, A. Aw, and H. Li. "A Linguistically Annotated Reordering Model for BTG-based Statistical Machine Translation." In *Proceedings of ACL 2008*.

[19] 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.

[20] 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.

[21] 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.

[22] B. Chen, M. Zhang, A. Aw and H. Li. "Regenerating Hypotheses for Statistical Machine Translation". In *Proceeding of COLING 2008*. Manchester, UK. Aug. 2008.

[23] 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.

[24] 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.

[25] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. "BLEU: a method for automatic evaluation of machine translation". In *Proceeding of ACL-2002*.