I²R's Machine Translation System for IWSLT 2010

Xiangyu Duan, Rafael E. Banchs, Jun Lang, Deyi Xiong, Aiti Aw, Min Zhang and Haizhou Li

Institute for Infocomm Research, Singapore
{xduan, rembanchs, jlang, dyxiong, aaiti, mzhang, hli}@i2r.a-star.edu.sg

Abstract

In this paper, we describe the system and approach used by Institute for Infocomm Research (I²R) for IWSLT 2010 spoken language translation evaluation campaign. We apply system combination on top of two kinds of statistical machine translation system, namely, phrase-based system and syntax-based system. Experimental results show consistent improvements on DIALOG Task.

1. Introduction

This paper describes the statistical machine translation (SMT) system and approach explored by Institute for Infocomm Research (I²R) for DIALOG Task of International Workshop on Spoken Language Translation (IWSLT) 2010.

The DIALOG Task targets at translating dialogs in travel situation bi-directionally, that is, from Chinese to English and from English to Chinese. We use the same strategy for both translation directions, with differences in details of data preprocessing, respective single system and system combination.

Basically, our strategy is to use a system combination framework on top of two kinds of statistical machine translation systems: phrase-based and syntax-based. These two kinds of systems provide different views of the translation process and parallel data structure, which enable system combination to explore the diversity of systems. We will present each individual system and system combination method respectively in section 2 and 3. Specific applications of them for each translation direction will be presented in detail in experimental section 4. Section 5 concludes the paper.

2. The SMT Models

To integrate the advantages of the state-of-the-art translation methods, we use two different SMT systems, phrase-based, and BTG-based systems. The two systems share some common features: word alignment of training data obtained from GIZA++[1], Language model(s) (LM) trained using SRILM toolkit [2] with modified Kneser-Ney smoothing method [3].

2.1. Phrasal Translation System: Lavender and Moses

Lavender [4] is our newly-developed in-house SMT translation platform, including a phrase-based decoder and most of the current linguistically motivated syntax-based system. Its phrase-based component, which functions very similar to Moses [5], is used as the phrase-based decoder for this campaign. Phrase-based SMT usually adopts a log-linear framework [6]. By introducing the hidden word alignment variable a [7], the optimal translation can be searched for based on the following criterion:

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

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 [8]. IBM word reordering constraints [9] are applied during decoding to reduce the computational complexity. The other models and feature functions employed by Lavender 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 [10] (RM)
- Word and phrase penalties, which count the numbers of words and phrases in the target string

Besides Lavender, we also utilize Moses to produce translations for system combination. Same feature functions are adopted in Moses [5].

2.2. Syntax-based Translation System: Tranyu

Tranyu is our other in-house translation platform. It is a formally syntax-based SMT system, which adapts the bracketing transduction grammars (BTG) for phrase translation and reordering. The BTG lexical rules (A --> xly) 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 in a log-linear form. For lexical rules, phrase/lexical translation probabilities in both directions, word/phrase penalties, as well as the language model are used as features. For merging rules, we incorporate maximum entropy (MaxEnt) based reordering models to predict orders between two neighboring phrases. We train all the model scaling factors on the development set to maximize the BLEU score. A CKY-style decoder is developed to generate the best BTG binary tree for each input sentence, which yields the best translation.

We developed three variations of Tranyu. Each variation was tuned independently on the development set. All variations share the same phrase table, language model and boundary word based reordering model. We give brief introductions of these variations as follows.

- Tranyu(Bound). In this variation, we use a boundary word based reordering (BWR) model [11] to predict phrase orders for merging rules. We define boundary words as words at the begining/ending positions of source/target sides of two neighboring phrases. Supposing the left phrase pair is "于 7月 15 日|on July 15", the right phrase pair is "举行 总统 与 国会 选举| held its

presidential and parliament elections", source words {“ 于”, “15 日”, “举行”, “选举” } and target words {“on”, “15”, “held”, “elections”} are boundary words. Training a BWR model proceeds through 3 steps. First, we extract reordering examples from word-aligned bilingual data, then generate features using boundary words of these examples and finally estimate feature weights.

- Tranyu(LAR). In order to employ more linguistic knowledge in the BTG reordering, we extend boundary word based reordering further by linguistically annotating each node involved in reordering according to the source-side parse tree. We call this linguistically annotated reordering (LAR). In LAR, we annotate each BTG node with three annotation elements: (1) head word, (2) the part-of-speech (POS) tag of head word and (3) syntactic category. We use these three elements, together with boundary words described above, as our reordering features. The weights of these features are tuned using a MaxEnt trainer. For more details, please refer to [12].

- Tranyu(UniBrack). Syntactic analysis influences the way in which the source sentence is translated. In this variation, except for the reordering model BWR, we incorporate a syntax-driven bracketing model (UniBrack) which predicts whether a phrase (a sequence of contiguous words) is bracketable or not using rich syntactic constraints. If a source phrase remains contiguous after translation, we refer this type of phrase {\bf bracketable}, otherwise {\bf unbracketable}. We parse the source language sentences in the word-aligned training corpus. According to the word alignments, we define bracketable and unbracketable instances. For each of these instances, we automatically extract relevant syntactic features from the source parse tree as bracketing evidences. Then we tune the weights of these features using a maximum entropy trainer. For more details, please refer to [13].

To further improve reordering between two neighboring phrases, we introduce two hard constraints. The first one is the swapping window, which only allows reordering within a pre-defined window (we set the window size to 15 words on the source side). The second one is the punctuation restriction, which prohibits any inverted orders if two neighboring phrases include any of the punctuation marks {， 、 ： ； 「 」 《 》 （ ） “ ”}. For more details, please refer to [14]. These two constraints are implemented in all three Tranyu variations described above.

3. System Combination

There are two methods which can be applied as a system combination module. One is rescoring method [15], which utilizes rich global features to re-rank n-best translations. For each input sentence, we concatenate translation hypotheses of each individual system into one n-best list and rescore it using global features. The 1best result obtained from rescoring is taken as final result. Rescoring is used as system combination module for English-to-Chinese translation. The other method is confusion network based system combination method [16]. Given hypotheses from all individual systems, we construct confusion network through hypothesis alignment and decode the best translation path from the constructed confusion network. This method is used as system combination module for Chinese-to-English translation task.

3.1. Rescoring

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 2009 [17]. We apply the following feature functions. Weights of feature functions are optimized by using 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 [18]
- 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
- Linear sum of n-grams (n=1,2,3,4) relative frequencies within all translations, which favors the hypotheses containing popular n-grams of higher order [19]
- n-gram posterior probabilities within the N-best translations [20]
- sentence length posterior probabilities [20]

3.2. Confusion Network Based System Combination

In general, confusion network based system combination consists of four steps: 1) Backbone selection: to select a backbone from all hypotheses. Backbone determines the word order of final translation. 2) Hypothesis alignment: to build word alignment between backbone and each hypothesis. 3) Confusion network construction: to build confusion network from hypothesis alignment. 4) Confusion network decoding: to find the best translation path through confusion network.

Among the four steps, hypothesis alignment presents the biggest challenge due to varying word orders between outputs from different machine translation systems. We apply four kinds of word alignment methods for hypothesis alignment:

- GIZA++ [1]. The alignments between backbone and hypothesis are obtained by using enhanced HMM model bootstrapped from IBM model-1. All hypotheses of test set are collected to create sentence pairs for GIZA++ training, which outputs many-to-1 word alignments.
- TER-based: The TER (translation error rate) score [21] measures the minimum number of string edits between hypothesis and reference (backbone in this case) where edits include insertions, deletions, substitutions and phrasal shifts. The best alignment is the one that gives minimum number of translation edits. TER produces 1-to-1 alignments.
- CLA-based: Competitive linking algorithm (CLA) [22] applies a greedy algorithm to search for word alignment with the highest sum of association score, which is computed on each word pairs between backbone and hypothesis. CLA produces 1-to-1 alignments.

- IHMM-based: Indirect hidden Markov model (IHMM) [23] estimates model parameters indirectly from various resources such as semantic similarity, distortion penalty. IHMM produces many-to-1 alignments.

4. Experiments

Experiments are conducted on DIALOG Task, which is about spoken dialog translation in travel situation between Chinese and English. The evaluation metric includes BLEU [24] and NIST score [25]. The translation input conditions of the DIALOG Task consist of: 1) automatic speech recognition (ASR) outputs, in which we choose 1best speech recognition results as inputs, and 2) correct recognition results (CRR), i.e., text input without speech recognition errors. Two subtasks are contained in DIALOG Task: English-to-Chinese and Chinese-to-English.

4.1. English-to-Chinese

4.1.1. Dataset description

The official training set comes from Spoken Language Databases (SLDB) corpus and parts of BTEC corpus which was released by IWSLT 2010 organizers. The official development set consisted of 4 datasets used in previous years' evaluations as development and test sets. These are namely: devset3 (from IWSLT05), devset10 and devset11 (from IWSLT08), and DIALOG devset. For development, we selected the first three of these datasets, for which seven translation references were available. For internal test, we used the last of the four provided official development datasets, for which four translation references were available. Table 1 gives the statistics of our data configuration.

Table 1: Number of sentences and available references in our training, development and internal test datasets.

	Train	Dev	test
No. of sent.	30,033	1,255	210
No. of refs.	-	7	4

4.1.2. Chinese segmentation and n-gram order

The first step towards the construction of our baseline system was to determine the type of segmentation to be used for Chinese and the n-gram order to be used for the language model. Four different segmentations types were considered: the original segmentation provided, automatic segmentations computed with ICTCLAS [26] and NUS tools [27], and character-based segmentation. Three different n-gram orders were also considered: 3, 4 and 5. Standard PBSMT systems were constructed by using the MOSES framework for the CRR version of the dataset and results were compared by means of the resulting BLEU[1] scores over our internal test set. Table 2 summarizes the results.

According to the results depicted in Table 2, we finally selected character-based segmentation for the Chinese side of the data and used a 5-gram language model as target language model for decoding. As an additional verification, a system using character-based segmentation and a 6-gram language model was constructed. The obtained BLEU score was 37.49.

Table 2: Comparative evaluation (in terms of translation BLEU) among different Chinese segmentation types and n-gram orders.

	3-gram	4-gram	5-gram
Original	37.89	38.13	38.31
ICTCLAS	35.46	35.95	37.04
NUS tool	37.50	35.64	36.81
Character	35.81	37.39	**38.38**

In the case of English (source side), standard tokenization and lowercasing was applied to the data.

4.1.3. Some additional considerations

Once the segmentation type and n-gram order have been determined we focused on any additional preprocessing details that might improve the system performance. The following are the most relevant ones we were able to identify:

- Training corpus enhancement: We included those non-overlapping development data portions from the Chinese-to-English translation task into our English-to-Chinese training data (namely: dev1, dev2, dev4, dev5, dev6, dev7, dev8 and dev9), which resulted in 3,941 new sentence pairs added to the training set.
- Elimination of heading and trailing blanks in the Chinese character segmentation. At some point we realized that provided tools for character segmentation introduced heading and trailing blanks in Chinese sentences, which had a substantial effect on GIZA alignments and BLEU scores. By removing those heading and trailing blanks we achieved an interesting system performance improvement.
- English hyphens removal. Some words in the English side of the dataset (such as twenty-five, rent-a-car, three-day) included hyphens. By removing hyphens from the English side of the data better alignments were obtained and a small improvement in performance was achieved too.
- Decoding parameters. System performance was additionally boosted up by adjusting some of the default settings for MOSES' decoding parameters. More specifically, stack size was increased to 1000, the translation table size limit was increased to 100, and Minimum Bayes Risk decoding was activated. Consequently, some translation quality was gained at the expense of decoding time increment.

Table 3: Additional considerations taken into account and their corresponding impact on translation quality.

System	BLEU
Baseline	38.38
+ Training corpus enhanced	40.88
+ Heading and trailing blanks eliminated	44.34
+ English hyphens removed	44.76
+ Decoding parameters adjusted	**45.53**

[1] All BLEU scores reported within the English-to-Chinese subsection are computed at the character level.

Table 3 summarizes the improvements achieved by including the aforementioned modifications to our baseline system.

4.1.4. ASR input processing

Experimental analysis and parameter adjustments presented in the previous subsections were conducted over the CRR version of the dataset. In addition to the strategies already defined for CRR data, the following considerations were taken into account for the ASR version of the dataset (we restricted our experimentation to the 1-best ASR output condition):

- Hesitations removed: a rule-based system, based on regular expressions, was developed for removing hesitations from the ASR dataset. The hesitations removed included expressions such as: *oh, ah, uh, hum, um, umm, mmm*, etc.
- Punctuation correction. For punctuation insertion, provided tools and guidelines were used. However, after visually inspecting punctuation insertion results, a significant number of errors were identified. According to this, a rule-based system was developed to attempt punctuation insertion correction. The implemented correction strategy was a very simple one: *If a question condition is met, then replace ending punctuation mark with "?"; otherwise, replace ending punctuation mark with "."*. Question conditions were identified by means of regular expression patterns such as: *(could|will|would|are|is|can|may|might|do|does|have|has|had) (you|it|either|i|we|that|there)*. In total, seven different of such patterns were designed and used.
- Word corrections. Finally, some basic rules for attempting to correct common ASR output errors were designed and implemented. The most relevant rule-based corrections implemented are:
    - *OK* corrections. Phonetically-inspired regular expressions were designed for replacing things such as *all k, all kay, oh kay*, etc. with *OK*.
    - Duplicate word elimination. A rule for eliminating duplicate words (with the exception of numbers!) was implemented.
    - Duplicate punctuation elimination. All sequences of more than one punctuation mark were reduced to the first punctuation mark in the sequence.
    - Phonetic corrections in number sequences. Phonetically-inspired rules were designed for replacing certain no-number words in sequences of the form *number no-number number*. Some examples of these rules included: *to* by *two* except if preceded by *from*, *and* by *one* except if followed by *yen*, *right* by *eight*, *of* by *o*, etc.

Table 4 summarizes the improvements achieved by applying the aforementioned preprocesses to the ASR input data.

Finally, an OOV removal strategy was implemented. According to this, all English words not translated into Chinese were removed form translation outputs. This allowed a further small increment in BLEU, from 45.53 to **45.55** and from 38.47 to **38.62**, for our CRR and ASR systems, respectively.

Table 4: Effects of ASR specific preprocessing on translation quality.

System	BLEU
Best CRR system evaluated on ASR	37.48
+ Hesitations removed	37.55
+ Punctuation correction applied	38.25
+ Word correction applied	**38.47**

4.1.5. System combination

Our system combination strategy for both English-to-Chinese input conditions: CRR and ASR, was based on the rescoring procedure previously described in section 3.1. Three systems were used in system combination: MOSES, Lavender and Tranyu(Bound). 500-best lists were generated for each system, and the resulting 1500 list of hypotheses was rescored by means of the global features described in section 3.1, plus three additional dummy binary features that were used to provide MERT and rescoring algorithms with discriminative information about the system-of-origin for each individual hypothesis.

Table 5 summarizes individual system and system combination performance over internal test set (in terms of translation BLEU) for both input conditions under consideration.

Table 5: Individual system and system combination performances for both input conditions: CRR and ASR.

System	ASR	CRR
MOSES	38.62	45.55
Tranyu(Bound)	37.39	44.01
Lavender	37.62	45.45
Combination	**38.75**	**45.98**

4.2. Chinese-to-English

4.2.1. Data description

The official training set comes from Spoken Language Databases (SLDB) corpus and parts of BTEC corpus. The official development set consists of 10 data sets used in previous years' development and evaluation. For internal test, we divide these 10 data sets into three parts: one part is added into official training set, one part is for internal development, one part is for internal test. Data split of these 10 development sets is as follows: set 1-7 are added into training set; set 9 is taken as internal development set, set 8 and 10 are combined as internal test set. Table 6 gives the statistics of this data configuration.

Table 6: Number of sentences in training set, internal development set and internal test set.

	train	Dev	test
No. of sent.	37,237	504	446

4.2.2. Data preprocessing

Data preprocessing is performed differently on each language side. On English side, tokenization and lowercasing are performed to reduce data sparseness. On Chinese side, different word segmentation, number detection and translation are ex-

plored. Table 7 presents the effect of preprocessing on Chinese side. The decoder in this study is Moses with GIZA++ word alignments.

Table 7: Test set performances (BLEU and NIST score) of data preprocessing on Chinese language side. "ORI" denotes original word segmentation, "ICT" denotes ICT's segmentation tool [26]. "NUS" denotes NUS's segmentation tool [27].

		BLEU	NIST
word segmentation	ORI	**0.4603**	**7.4042**
	ICT	0.3956	6.5231
	NUS	0.4038	6.5812
ORI + number translation		0.4587	7.3248

Through the comparison on different Chinese word segmentation, we can see that original word segmentation significantly outperforms other word segmentation tools. The reason locates in long numbers. For example, one sentence containing long number in original segmentation is: "电话 二零二七八二一二一六", while other word segmentation tool will segment the long number into one word: "电话 二零二七八二一二一六", which causes severe data sparseness.

One solution is to translate such numbers in advance and then let the Moses decoder to compete such translations with other translation options. Row "number translation" shows the performance of such solution, which adopts original word segmentation. The performance is similar to not doing so. Finally, we decide to use original word segmentation and without using number translation in the following experiments.

4.2.3. *Word alignment combination*

Most phrase-based and syntax-based systems extract translation rules from word alignments. The extracted translation rules will be enriched if multiple kinds of word alignments are concatenated together. There are several widely-used heuristics for word alignment to balance precision and recall. Each heuristics will generate one word alignments. We combine word alignments from different heuristics, extract phrase table from such word alignments, and test the performance by using phrase-based decoder: Moses.

Table 8: Performance (BLEU and NIST score) of word alignment combination

		BLEU	NIST
GIZA++	baseline	0.4603	7.2618
	combination	**0.4749**	7.3573
Berkeley	baseline	0.4608	7.1290
	combination	0.4717	**7.4001**

Two word alignment tools are applied: GIZA++ [1] and Berkeley alignment tool [28]. For GIZA++, baseline heuristics is "grow-diag-final-and"; for Berkeley alignments, baseline heuristics is "grow". Table 8 shows that word alignment combination improves the performance. We adopt GIZA++'s word alignments combination for the following Chinese-to-English experiments.

4.2.4. *Rescoring*

In Chinese-to-English translation, each system uses rescoring individually to re-rank its n-best outputs. This is a big difference to English-to-Chinese translation, which uses rescoring as system combination method.

Through rescoring, each system can provide more qualified n-bests for the following system combination. Totally, we applied rescoring on four systems, namely, Moses and three variations of Tranyu. The performances are shown in Table 9. Row "before" reports performance before rescoring while row "after" reports performance after rescoring.

Table 9: Rescoring performances (BLEU) on test set.

	Moses	Tranyu: Bound	Tranyu: UniBrack	Tranyu: LAR
before	0.4749	0.4719	0.4726	0.4685
after	0.4899	0.4954	0.4794	0.4845

Rescoring improves performance on Moses, Tranyu-Bound and Tranyu-Lar, while improves marginally on Tranyu-UniBrack.

4.2.5. *Confusion network based system combination*

Table 10 presents the performance of confusion network based system combination, which utilize four systems in total, namely, Moses, Tranyu-Bound, Tranyu-Unibrack and Tranyu-Lar. We can see that confusion network based system combination improves the performance over best single system: Tranyu-Bound (after rescoring), and the best performance is obtained by using GIZA++ for hypothesis alignment.

Table 10: Performance (BLEU) of confusion network based system combination.

Tranyu-Bound	Confusion Network			
	GIZA++	TER	CLA	IHMM
0.4954	**0.5054**	0.5020	0.5020	0.4995

4.2.6. *Experiments on ASR input*

There are two special preprocessing measures for ASR input: punctuation insertion and word-to-Pinyin conversion.

We use SRILM tool to perform punctuation insertion to restore punctuations missing in ASR output. We also perform word-to-Pinyin conversion to address the problem that ASR input contains word errors from automatic speech recognition. Some of the word errors can be bypassed by converting words into Pinyin because different words of ASR may share the same Pinyin. Table 11 reports experimental results of punctuation insertion and Pinyin conversion.

Table 11: Performances (BLEU and NIST score) of punctuation insertion and Pinyin conversion on ASR input.

	BLEU	NIST
word	0.3615	5.8342
word+punc.	0.3738	6.2582
Pinyin	0.3730	5.9933
Pinyin+punc.	**0.3899**	**6.3262**
char. Pinyin	0.3674	6.2867
char. Pinyin+punc.	0.3718	6.1883

Row "word" shows the baseline performance of ASR input. Row "Pinyin" shows that word-to-Pinyin is effective by improving BLEU from 0.3615 to 0.3730. Row "char Pinyin" denotes that word Pinyin is split into character Pinyin, which shows marginal improvement over baseline. All rows about "*+punc" denotes the effects of punctuation insertion, which consistently improves the performance.

5. Conclusion

This paper describes I^{2}R's SMT system that is used in the DIALOG Task of IWSLT 2010 MT campaign. We use a system combination framework that incorporates mainly two kinds of our SMT systems: phrase-based and syntax-based systems in the IWSLT 2010. We explain the details of our experiments and report how we achieve the final performance from single systems to the combined systems step by step.

6. References

[1] 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.

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

[3] 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.

[4] Min Zhang et al., "Lavender: I2R Statistical Machine Translation Platform", Technical-report-2009-008, Institute for Infocomm Research, 2009.

[5] 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.

[6] F. J. Och, and H. Ney. "Discriminative Training and Maximum Entropy Models for Statistical Machine Translation." In *Proceeding of ACL-2002*. 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." In *Proceedings of ACL-2003*. Sapporo, Japan. 2003.

[9] 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.

[10] 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*.

[11] Deyi Xiong, Qun Liu, and Shouxun Lin. "Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation". In *Proceedings of COLING-ACL 2006*, Sydney, Australia.

[12] Deyi Xiong, Min Zhang, Ai Ti Aw, and Haizhou Li. "A Linguistically Annotated Reordering Model for BTG-based Statistical Machine Translation". In *Proceedings of ACL 2008*.

[13] Deyi Xiong, Min Zhang, Aiti Aw and Haizhou Li. "A Syntax-Driven Bracketing Model for Phrase-Based Translation". In *Proceedings of ACL-IJCNLP 2009*.

[14] Deyi Xiong, Min Zhang, Ai Ti Aw, Haitao Mi, Qun Liu and Shouxun Lin. "Refinements in BTG-based Statistical Machine Translation". In *Proceedings of IJCNLP 2008*.

[15] B. Chen, Jun Sun, Hongfei Jiang, Min Zhang and Aiti Aw. "I2R Chinese-English Translation System for IWSLT-2007", In *Proceeding of IWSLT 2007. pp. 55-60. October Trento, Italy,* 2007.

[16] B. Chen, Min Zhang, Haizhou Li and Aiti Aw. "A comparative study of hypothesis alignment and its improvement for machine translation system combination", In *Proceedings of ACL-IJCNLP 2009*.

[17] Xiangyu Duan, Deyi Xiong, Hui Zhang, Min Zhang and Haizhou Li "*I2R's Machine Translation System for IWSLT 2009*" In *IWSLT, Tokyo, Japan, December* 2009.

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

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

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

[21] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul. "A study of translation edit rate with targeted human annotation". In *Proceeding of AMTA*, 2006.

[22] I. D. Melamed. "Models of translational equivalence among words". *Computational Linguistics, 26(2), pp. 221-249*, 2000.

[23] X. He, M. Yang, J. Gao, P. Nguyen, R. Moore. "Indirect-HMM-based Hypothesis Alignment for Combining Outputs from Machine Translation Systems". In *Proceeding of EMNLP. Hawaii, US*, Oct, 2008.

[24] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. "BLEU: a method for automatic evaluation of machine translation". In *Proceeding of ACL-2002, pp. 311-318*, 2002.

[25] NIST Report. "Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics". *http://www.nist.gov/speech/tests/mt/doc/ngramstudy,*2002.

[26] Huaping Zhang, Hongkui Yu, Deyi Xiong, and Qun Liu. "HHMM-based Chinese lexical analyzer ICTCLAS". In *Proceedings of Second SIGHAN Workshop on Chinese Language Processing, pages 184–187, Sappora, Japan*. 2003.

[27] Jin Kiat Low, Hwee Tou Ng and Wenyuan Guo , "A Maximum Entropy Approach to Chinese Word Segmentation". In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, 2005.

[28] P. Liang, B. Taskar and D. Klein, "Alignment by Agreement", In *Proceedings of North American Association for Computational Linguistics (NAACL)*, 2006.