The *ICT*'s Patent MT System Description for NTCIR-9

Hao Xiong
Key Lab. of Intelligent Information Processing
Institute of Computing Technology, CAS
P.O. Box 2704, Beijing 100190, China
xionghao@ict.ac.cn

Linfeng Song
Key Lab. of Intelligent Information Processing
Institute of Computing Technology, CAS
P.O. Box 2704, Beijing 100190, China
songlinfeng@ict.ac.cn

FanDong Meng
Key Lab. of Intelligent Information Processing
Institute of Computing Technology, CAS
P.O. Box 2704, Beijing 100190, China
mengfandong@ict.ac.cn

Yajuan Lü
Key Lab. of Intelligent Information Processing
Institute of Computing Technology, CAS
P.O. Box 2704, Beijing 100190, China
lvyanjuan@ict.ac.cn

Qun Liu
Key Lab. of Intelligent Information Processing
Institute of Computing Technology, CAS
P.O. Box 2704, Beijing 100190, China
liuqun@ict.ac.cn

ABSTRACT

This paper introduces the ICT's system for Patent Machine Translation at the NTCIR-9 Workshop. In this year's program, we participate all the three subtasks: Chinese-English, English-Japanese and Japanese-English. We submit six translation results for each subtask generated by an in-house implemented hierarchical phrase-based system (HPB) with four different variants, a widely used open source system (Moses) as well as a combinational system (SCM), respectively. We employ general translation model and concentrate on developing refined preprocessing and postprocessing techniques for patent translation. Besides that, we attempt to improve the quality of patent translation by chemical expression substitution, incorporating manually written templates, domain adaption and reranking, etc. Experimental results show that our small techniques achieve improvement over baseline, however, compared to other participants, our result is not excellent.

Categories and Subject Descriptors

D.2.8 [**Natural Language Processing**]: Machine Learning—*Machine Translation, Patent Translation, Boosting*

General Terms

Translation

Keywords

Machine Translation, Patent Translation, System Combination
Team Name: [ICT]
Subtasks/Languages: [Chinese-to-English, Japanese-to-English and English-to-Japanese]
External Resources Used: [Chasen, ICTCLAS, SRI, Giza++, Moses]

1. INTRODUCTION

This year's Patent Machine Translation task[2] at the NTCIR-9 workshop consists of three subtasks: Chinese-English, English-Japanese and Japanese-English. We participate all subtasks and submit six system results for each subtask. Our submissions are mainly generated by three traditional machine translation models: phrase-based translation model[4], hierarchical phrase-based model[1] and system combinational model[9]. The first submission is generated by combinational system, the second to fifth are produced by in-house implemented hierarchical phrase-based model with four different rule filter strategy, and the last one is from well known open source toolkit: Moses[3].

The reason why we choose three traditional models for this year's campaign is that patent documents mainly consists of introductive and descriptive sentences, while our syntactic parser is trained on news corpus that performed badly on patent documents. Additionally, we found that the phrase-based model performs surprisingly very well on patent documents. For that reason, we omit complex tree-based model[5, 8] which we have obtained promising results in previous NIST MT evaluations, mainly concentrate on exploring refined techniques for tokenization, segmentation as well as alignment while decoding with frequently-used hierarchical SMT model.

Specifically, we modify our segmenter to generate better segmented results for patent sentences, and incorporate manually written translation templates into the decoder to improve the translation result. In virtue of provided corpus contains mixed sentences from chemical, physical and medical, etc., domains, we design an approach to classify them into identical categories and translate them independently. Moreover, we substitute chemical expression into special characters own to difficulty in segmentation, tokenization and alignment for these contents. Experimental results show that some of our approaches gain improvement in term of BLEU score while some doesn't. However, we will still introduce these methods in the rest of this paper.

The remainder of this paper is organized as follows, we illustrate overall system architecture along with detailed tech-

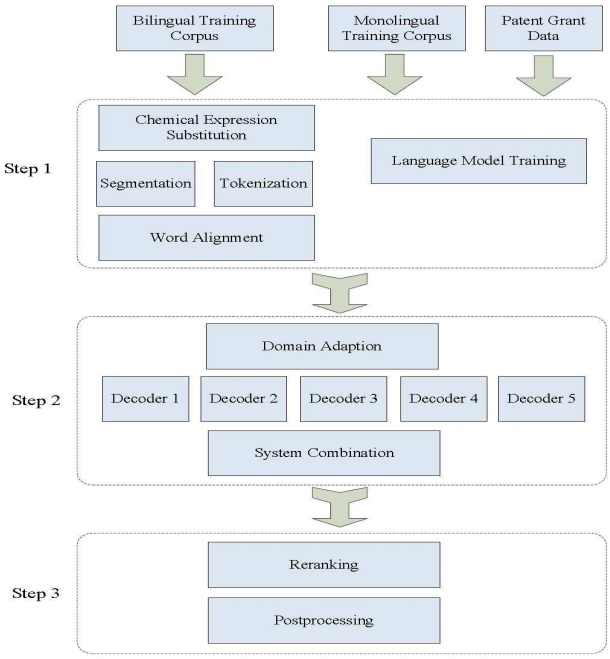

Figure 1: The overall architecture of our system, where step 1 is the corpus preprocessing stage preparing for the step 2, in which corpus are divided into several domains and then further used for independent decoding. Step 3 involves reranking techniques for each submission and final postprocessing procedure.

nical descriptions in Section 2. Section 3 mainly presents our experimental results, and we conclude our paper in Section 4.

2. SYSTEM ARCHITECTURE

We illustrate system architecture in figure 1, where the whole process includes three steps including corpus preprocessing, translation generating and result postprocessing, respectively. Most of our techniques in this architecture are similar to the Moses toolkit's [1], thus in this section, we will mainly focus on describing the special designed parts of our system while temporarily omit other skills which will be later introduced in the experimental section.

2.1 Chemical Expression Substitution

According to our statistics, almost sixth training corpus stems from chemical domain, and a certain amount of these sentences contains chemical expressions (Some medical documents also contains kinds of chemical expressions). Generally, chemical expression is comprised of numbers, terminologies and punctuation, while classical statistical-based segmenter and tokenizer are tend to split it into several parts due to seldom occurrence in the training corpus. Furthermore, divided punctuation and numbers in chemical expression will be incorrectly aligned in case another similar counterparts exist in that sentence. In such case, error will be propagated and result in poor translation of chemical sentences. Therefore, before tokenization and segmentation, we use manually written rules to recognize chemical expression in the training corpus and substitute it with a special symbol that doesn't appear in the corpus elsewhere. Noting that, we are unfamiliar with Japanese, hence we perform such substitution only on Chinese and English corpus.

Here, we just introduce the procedure of recognition on Chinese side, since recognition on English side has the same procedure. Firstly, one dictionary comprising chemical terminologies, is prepared to recognize the location of chemical expressions in Chinese corpus. We name it as location dictionary. To recognize chemical expressions, we design following rules:

1. If a word in one sentence is found in the location dictionary, the word may be a part of a chemical expression. And the next step is to determine the boundary of this expression.

2. If the character before the current location is "-", and the word after "-" is a Chinese number like "二', then n (n equals the Chinese number: "2" equals '二') Arabic numbers separated by commas should be found before "-".

3. If the character before the current location is "-", and the word after "-" is not a Chinese number, one Arabic number is needed before that.

4. Special letter like Greek letter appears next to the above boundary characters is also viewed as part of a chemical expression.

For example, to recognize chemical expression in the sentence"向多颈烧瓶中投入1,6-二氢-6-氧代-4-嗪啶羧酸。", we first identify three location words "二氢", "氧代", "嗪啶羧酸" using location dictionary. Then, since the character after the first "-" is "二", two Arabic numbers before it are recognized as parts of this chemical expression. With the same procedure, we will recognize the whole chemical expression "1,6-二氢-6-氧代-4-嗪啶羧酸".

It is worth noting that we substitute chemical expressions before segmentation and word alignment until rule extraction step. The reason to do that is mentioned in former sections. However, such process only intuitively guarantee to improve the segmentation and alignment for patent documents. Nevertheless, to translation, is still useless, since it is almost impossible to find a similar chemical expression in the training corpus for testing instance. Therefore, for each recognized Chinese chemical expressions, we capture its counterparts in the English side, and then collect them to build a dictionary which will be further cumulated to bilingual sentence pairs for word alignment. Surely, those bilingual chemical expression pairs will also be used to extract rules directing final translation.

2.2 Refined Segmentation

In previous Chinese-to-English machine translation evaluation campaigns, most participants utilized *ICTCLAS*[2] as their segmenter. To our knowledge, *ICTCLAS* is trained on news area which lacks of terminologies from patent documents. Thus, terminologies or chemical expressions last

[1] http://www.statmt.org/moses_steps.html

[2] http://ictclas.org/ictclas_feedback.aspx?packetid=49&packeturl=down/50/ICTCLAS50_Windows_32_C.rar

section mentioned will possibly be incorrectly segmented. It is worth mentioning that in last section we just deal with chemical expressions while other specialized characters like formulas, medical terminologies, are directly segmented by segmenter. One solution to address this problem is to disambiguate segmentation by incorporating a terminological dictionary. Due to the hardness of obtaining the source of *ICTCLAS* and no right to modify its function, as an alternative, we implement a perceptron based segmenter and make some modifications towards patent documents.

We train our segmenter on the merging corpora of People's Daily (PD) corpus and Microsoft Research (MSR) corpus by average perceptron learning algorithm. In addition, some relevant rules about numerals and string processing are added to this tool to better handle named entities. It also allows users to add dictionary by themselves. This function is realized by the following steps. Firstly, for a given sentence, we find out all the words which are also in the dictionary, and store them in a queue in order. Secondly, when a word is in the dictionary queue, it will be re-weighted during decoding procedure. This tries to ensure words in the dictionary can be segmented correctly.

2.3 Domain Adaption

In the previous section we mentioned that the provided corpus includes kinds of documents from different domains. Intuitively, splitting the mixed corpus into identical domains should better translate them well. Benefit from other projects, [3] we have classified corpus from chemical, medical, mechanical and traditional Chinese medical domains. Therefore, we could perform supervised classification to achieve better precision.

We use general naive Bayes classification algorithm [6] in that its simple and excellent performance in previous work. For a new sentence s, we use the formula $\tilde{c} = \arg\max_{c \in C} P(c|s)$, where C includes aforementioned four domains, to assign it to the class c with the highest predicted probability. The overall process of domain adaption is presented as follows:

- Classify training corpus into identical domains.
- Classify developing corpus into identical domains.
- Tune weights independently for each domain.
- Classify testing corpus into identical domains.
- Translate testing sentence using corpus from its corresponding domain.

2.4 Incorporating Translation Templates

Typically, there are lots of sentences with relatively fixed translation pattern in patent documents. It would be beneficial for SMT systems to generate better translation if we manually write translation templates for these patterns and incorporate them into decoder. Based on this intuition, we write translation patterns manually only according to the language phenomenon occurring in training set.

During the decoding phase, these manual written translation patterns are utilized via the following manner: first match the input sequence with the source side of every translation patterns. If there are matched patterns in a certain span, additional hypotheses associated with these patterns are generated for this span. Note that these additional hypotheses will coexist with those hypotheses produced by applying traditional hierarchical phrasal rules.

Take the Chinese sentence "均由 导线 并联到 主电路 上" as an example. If there is a manual written translation pattern "均由##1并联到##2上 → are all by ##1 connected to ##2 in parallel ", which has two variables. Through pattern matching, we can find that the example pattern covers the whole sentence. When the decoder is computing the hypotheses for the span covering the whole sentence, besides the hypotheses produced by applying the hierarchical phrasal rules, some extra hypotheses will be added by combining the right-hand-side of the above pattern and the hypotheses for "导线" and "主电路".

2.5 Variants of Decoder

Since the number of rules extracted from hierarchical phrase-based model is too huge, we hence propose a simple method to reduce the size of rule table. We filter the rule whose $frac$ score is lower than threshold 0.0, 0.9, 1.0, 1.1, and obtain four decoders using different rule tables. The $frac$ score has the form similar to frequent count, and is mainly inspired by the literature [7]. However, large-scale experiments show that rule filter technique is unstable, we thus submit all results of different variants using threshold 0.0, 0.9, 1.0 and 1.1.

2.6 Multi-system Reranking

We also propose a bagging-based multi-system reranking technique to improve the quality of mixed multi-domain patent translation.

As for training, we bootstrap N new development sets from the original set, then we tune a subsystem using each newly generated set. In the decoding stage, for each sentence, we first decode it using all subsystems and generate a k-best candidate list from each subsystem. After that, we fuse these k-best list and eliminate similar deductions. Finally, we rerank the integrated k-best list by the sum of voting score from each subsystem. The voting score is the dot product of the relative candidate's feature vector and the relative subsystem's weight vector. For more details, readers can refer to [10].

3. EXPERIMENTS

3.1 Data Usage

The organizers provide both bilingual patent description sentence pairs for each subtasks as well as monolingual patent grant documents for Japanese and English. However, provided monolingual corpus contain large-scale sentences which is beyond our ability to handel them, thus we just take some portion of them to train language model. The overall corpus we use in our system is presented in table 1.

3.2 Preprocessing of Japanese

Japanese is a kind of agglutinative languages. Its biggest characteristic is to indicate the grammatical relations in a sentence by means of adhering function words to behind of notional words. There are no obvious boundaries between words in Japanese. So, Japanese word segmentation is one of necessary procedures on machine translation of Japanese-to-English. Preprocessing on Japanese corpus in out task

[3]Due to contract provision, here we can't announce the source and details of this patent corpus.

System	Bilingual	Monolingual
C-E	1 Million	40 Million
J-E	3 Million	40 Million
E-J	3 Million	73 Million

Table 1: The overall corpus we used in our system, wherein monolingual corpus are used to train language model.

consists of two procedures.

1. Full-width characters converting to half-width ones: In computer editorial process, letters, numbers and symbols may appear in half-width or full-with forms. This phenomenon will affect phrases' identification in the translation process in some extent, which may reduce the translation quality in the end. So we converted full-width characters to half-width ones in corpus in the first step.

2. Japanese word segmentation: Japanese word segmentation is basic task of Japanese information processing, which is also the foundation of Japanese machine translation. We used Chasen (chasen-2.4.4)[4], one of the most famous open source Japanese lexical analysis tools, to do the task of Japanese word segmentation in the second step. Chasen is developed by Nara Institute of Science and Technology, which is based on Hidden markov model.

3.3 Baseline Systems

First, for each subtask we first build one baseline system with preprocessing techniques similar to Moses's . Table 2 show performance of our different decoders running on three subtasks. Compared to phrase-based model Moses, the performance of our hierarchical phrase-based model is slightly higher. One reason is that hierarchical phrase-based model can reorder phrases between high distance while phrase-based model could generally explore local reordering. Another funding is that our combinational system achieves improvement over single system particularly on J-E direction. Although system combination techniques perform unstably when given few single inputs, however, in our experiments, it works well and the result is surprisingly positive.

System	C-E	J-E	E-J
HPB	30.08	23.55	32.85
Moses	29.56	23.31	32.27
SCM	**30.73**	**25.29**	**33.50**

Table 2: Experimental results of our decoders on three subtasks, where HPM is our hierarchical phrase-based model and SCM denotes system using combination technique.

3.4 Experiments of Language Model

We use the SRI Language Modeling Toolkit [11] to train the Japenese/English 5-gram, 6-gram, 7-gram language model with Kneser-Ney smoothing on the Japenese/English side of the training corpus respectively. Noting that, here we only use the monolingual portion of bilingual corpus. Table 3 gives the experimental results using different n-gram language models.

[4] `http://chasen-legacy.sourceforge.jp/`

Table 3: Experiments of different n-gram language models.

Task	System	5-gram LM	6-gram	7-gram
C-E	HPB	30.08	**30.34**	30.05
	Moses	**29.56**	28.93	29.05
J-E	HPB	23.55	23.15	**24.37**
	Moses	23.31	23.13	**23.55**
E-J	HPB	32.85	**32.90**	32.57
	Moses	32.27	32.10	**32.45**

From table 3, it is hard to predict which gram of language model will achieve the best performance on the final test. One acceptable explanation is that when gram of language model increases, data spareness problem will become more serious and results in substantive backoff.

Remember that the organizers applied additional monolingual corpus, we try our best to exploit all of them, however, we could only use small proportions from 2003 to 2005 to train another 5-gram language model. From table 4 we delightedly find that large-scale language model largely improve the performance of our system. But to our computational ability, we could exploit some of them. We believe the performance of our system will be further improved when using larger language model.

System	C-E	J-E	E-J
HPB	31.28	25.27	34.04
Moses	30.78	25.28	33.72
SCM	**32.67**	**25.63**	**34.55**

Table 4: Experimental results of our decoders on large-scale language model. We use two language model, one is trained on monolingual portion of bilingual description corpus while another is trained on monolingual patent grant corpus.

3.5 Threshold of Rule Filter

We have mentioned in last section that we use a threshold p to control the size of rule table for hierarchical phrase-based model. Table 5 is the experimental results of different variants. From table 5, we find that when setting threshold p to 1.1, the system obtains comparable higher BLEU score than others. The reason is that rule filter can drop some rules that are incorrectly extracted due to incorrect word alignment, however we can't determine which threshold could always achieve the best performance. Since the organizers require us marking order of each submission, we thus mark the order as 1.1>1.0>0.9>0.0 based on experimental results on developing set. The feedback score of final submission also supports the right of our decision.

3.6 Final Results

System	C-E	J-E	E-J
$p = 0.0$	31.28	25.27	**34.04**
$p = 0.9$	**31.72**	25.66	33.45
$p = 1.0$	31.67	25.46	33.42
$p = 1.1$	31.51	**25.86**	33.51

Table 5: Experimental results of different threshold for rule filter.

In this subsection, we present our system results on final testing set. We submit six systems for each subtasks, labeled as:

- *sys1*:combinational system trained on following five single systems.
- *sys2*:hierarachical phrase-based model with rule filter threshold $p = 1.1$.
- *sys3*:hierarachical phrase-based model with rule filter threshold $p = 1.0$.
- *sys4*:hierarachical phrase-based model with rule filter threshold $p = 0.9$.
- *sys5*:hierarachical phrase-based model with rule filter threshold $p = 0.0$.
- *sys6*:Moses

System	C-E	J-E	E-J
sys1	**31.97**	**27.28**	**32.91**
sys2	31.52	26.90	32.10
sys3	31.57	26.55	31.72
sys4	30.78	26.71	32.06
sys5	30.76	26.06	32.17
sys6	30.64	26.84	30.17

Table 6: Evaluation results of our final submission.

Table 6 lists the results of our final submission, where sys1 to sys6 are defined in above itemization. Although we attempt several techniques during the campaign, but most of them using outer corpus, thus we submit the final result only using limited corpus, and show extended experiments in next subsection.

3.7 Extended Experimental Results

In this subsection, we will present experimental results which we attempted during evaluation task but didn't submit in the final test.

3.7.1 Incorporating Templates

We use about 20 thousand manually written templates to improve the performance of system on C-E subtask. However, from table 7, we find the BLEU score apparently drops. Since our manually written templates are from another project, the target translation of written template differs from the translation of this year's requirements. Moreover, we use only one reference for evaluating the translation quality, it is hard for one decoder with manually written rules generates the same translation as referential results. To this respect, BLEU score is better for statistical machine translation model but not better for rule-based translation model.

System	C-E
baseline	**30.08**
template-based	29.39

Table 7: Experimental results of incorporating manually written templates.

3.7.2 Chemical Expression Substitution

To evaluate the contribution of chemical expression substitution technique, we first substitute all available expressions both in bilingual and monolingual corpus, and we then heuristically extract aligned chemical expressions from bilingual sentence pairs. Using these bilingual chemical expression pairs, we could extract fine-grained rule which is helpful for translating terminologies. According to our statistics, we extract almost ten thousand pairs from bilingual documents. As the table 9 shows, although few chemical expressions are found in testing set, the performance of this technique is still slightly higher than the baseline. We argue that if more chemical expressions appear in the testing set, the performance will be further improved.

System	C-E
baseline	30.08
chemical expression substitution	**31.19**

Table 8: The results of using chemical expression substitution.

3.7.3 Refined Segmentation

Since our segmenter is trained on non-patent corpus, its segmented results of patent documents is not very well particularly on chemical sentences. We thus incorporate an outer dictionary from other project to better the segmentation. However, results in table 9 strikes our intuition. The reason is that we just use dictionary to better the segmentation and further improve the word alignment, whereas, coarse phrase might cause spareness in translation while we doesn't handle it like chemical expression substitution in which we cumulate chemical expression dictionary to extract rules and generate fine-grained translation rules.

System	C-E
baseline	**30.08**
refined segmenter	29.39

Table 9: Experiments of refined segmentation.

3.7.4 Domain Adaption

Using outer classified corpus, we supervised divides training corpus into four domains. Table 10 lists the statistics of each domain, where the sentence from medicine appear frequently in the given data. Using classified training corpus,

Domain	C-E
Chinese Traditional Medicine	37124
Chemical	159492
Physical	529441
Medical	304928

Table 10: Sentence pairs of different domains in training corpus.

we could divide developing and testing sentences into corresponding domains which yields no more than 300 sentences of each domain for developing set. Experientially, tuning weights on such few sentences generally causes over-fitting problems in the testing set. Conversely, as the table 11 shows, our method improves the performance slightly over the baseline.

System	C-E
baseline	30.08
domain adaption	**30.22**

Table 11: Experimental result of domain adaption.

3.7.5 Multi-documents Reranking

We also evaluate our bagging-based reranking method on C-E direction. We generate one new developing set via randomly selecting 1000 sentences from original developing corpus, and repeat 30 times. Noting that, there are some reduplicative sentences in new generated developing set, since original developing corpus contains only 1000 sentences. Table 12 presents the results of reranking technique. We are glad to find that our method significantly outperform the baseline, the reason is given in [10].

System	C-E
baseline	31.08
reranking	**31.90**

Table 12: Result of multi-documents reranking. (This baseline is different from previous experiments, since it use different language model.)

4. CONCLUSION

In this paper, we summarize techniques we used in this year's evaluation tasks. We participate three subtasks of Patent Machine Translation task and submit six systems for each subtask. Also, we attempt several methods towards improving the quality of patent translation including refined segmentation, chemical expression substitution, domain adaption and multi-documents reranking, etc.

However, compared to other participants, our final result is not very competitive particularly on J-E direction. Based on this year's experience of patent translation, in next year's task, we are willing to concentrate on developing novel model for patent translation, elaborate techniques for patent data preprocessing and postprocessing.

5. ACKNOWLEDGMENTS

First, we thank respected organizers for their tireless and hard preparation. And we thank reviewers for their helpful comments. We also thank our colleagues who gave us endless help in the period of evaluation task. Lastly, we want to thank ShuGuang and ShenTeng super computing center, thank them for supporting us valuable computing resources.

This work was funded by National Natural Science Foundation of China, Contracts 60736014 and 60873167.

6. ADDITIONAL AUTHORS

Additional authors: Jun Xie (email: `xiejun@ict.ac.cn`) and Hui Yu (email: `yuhui@ict.ac.cn`)and Wei Luo (email: `luowei@ict.ac.cn`) and Miao Yu (email: `yumiao@ict.ac.cn`)

7. REFERENCES

[1] D. Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics, 2005.

[2] K. P. C. E. S. Isao Goto, Bin Lu and B. K. Tsou. Overview of the patent machine translation task at the ntcir-9 workshop. In *Proceedings of the NTCIR-9 Workshop*.

[3] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, et al. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics, 2007.

[4] P. Koehn, F. Och, and D. Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics, 2003.

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

[6] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer, 1998.

[7] H. Mi and L. Huang. Forest-based translation rule extraction. In *Proceedings of EMNLP*, pages 206–214, Honolulu, Hawaii, October 2008.

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

[9] A. Rosti, S. Matsoukas, and R. Schwartz. Improved word-level system combination for machine translation. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 312, 2007.

[10] L. Song, H. Mi, Y. Lv, and Q. Liu. Bagging-based system combination for domain adaption. In *Machine Translation Summit XIII*, 2011.

[11] A. Stolcke. Srilm - an extensible language modeling toolkit. In *Proceedings of ICSLP*, volume 30, pages 901–904, 2002.