NTT-UT Statistical Machine Translation in NTCIR-9 PatentMT

Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, Masaaki Nagata
NTT Communication Science Laboratories
2-4 Hikaridai, Seika-cho, Kyoto, Japan
{sudoh.katsuhito,kevin.duh,hajime.tsukada,masaaki.nagata}@lab.ntt.co.jp

Xianchao Wu[*], Takuya Matsuzaki, Jun'ichi Tsujii[†]
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
wu.xianchao@lab.ntt.co.jp, matuzaki@is.s.u-tokyo.ac.jp, jtsujii@microsoft.com

ABSTRACT

This paper describes details of the NTT-UT system in NTCIR-9 PatentMT task. One of its key technology is system combination; the final translation hypotheses are chosen from n-bests by different SMT systems in a Minimum Bayes Risk (MBR) manner. Each SMT system includes different technology: syntactic pre-ordering, forest-to-string translation, and using external resources for domain adaptation and target language modeling.

Categories and Subject Descriptors

I.2.7 [**Natural Language Processing**]: Machine Translation

General Terms

Algorithms, Performance

Keywords

pre-ordering, Generalized Minimum Bayes Risk (GMBR) system combination, Bayesian word alignment adaptation

Team Name

[NTT-UT]

Subtasks/Languages

[English-to-Japanese][Japanese-to-English][Chinese-to-English]

External Resources Used:

[Moses][MGIZA++][Enju][Cabocha][SVM^{rank}][NIST OpenMT 2008 Chinese-to-English]

1. INTRODUCTION

Statistical machine translation (SMT) is a promising way for machine translation in domains in which large-scale bilingual language resources are available. The NTCIR-9 PatentMT task [5] provided millions of parallel sentences for translating patent documents in three language pairs: Chinese-to-English, Japanese-to-English, English-to-Japanese. The NTT-UT team (organized by NTT and University of Tokyo) participated all of the three translation tracks with SMT systems.

One of the key technology of the NTT-UT system is "system combination" of phrase-based SMT with different conditions of training, preprocessing, and decoding algorithms, based on Generalized Minimum Bayes Risk (GMBR) [4]. Each individual system has a different aspect depending on its language pair. Other key technologies include syntactic pre-ordering, forest-to-string translation, and domain adaptation. In English-to-Japanese, we employed HPSG parsing for syntactic pre-ordering and forest-to-string translation. The pre-ordering is based on Head-Finalization [8], which moves syntactic heads toward the end of their siblings to reorder English words in Japanese-like word order. In Japanese-to-English, we employed dependency parsing for rule-based pre-ordering, in which Japanese chunks (called *bunsetsu*) are reordered based on their syntactic roles determined by dependency relation and function words. In Chinese-to-English, we utilized external bilingual resources in another domain using our domain adaptation technology.

The remainder of this paper is organized as follows. Section 2 presents our general framework of GMBR system combination. Section 3 to 5 describe our systems for each language pair tracks and their resuts: English-to-Japanese (Section 3), Japanese-to-English (Section 4), and Chinese-to-English (Section 5). Finally, Section 6 discusses our conclusions.

2. GENERALIZED MINIMUM BAYES RISK SYSTEM COMBINATION

This section briefly present our GMBR system combination. Please refer to our paper [4] for details. Note that our system combination only picks one hypothesis from an N-best list and does not generate a new hypothesis by mixing partial hypotheses among the N-best.

2.1 Theory

Minimum Bayes Risk (MBR) is a decision rule to choose hypotheses that minimize the expected loss (i.e. *Bayes Risk*).

[*] Currently with NTT Communication Science Laboratories (since April 2011).

[†] Currently with Microsoft Research Asia (since May 2011).

In the task of SMT from a French sentence (f) to an English sentence (e), MBR decision rule on $\delta(f) \rightarrow e'$ with the loss function L over the possible space of sentence pairs $(p(e,f))$ is denoted as:

$$\operatorname*{argmin}_{\delta(f)} \sum_{e} L(\delta(f)|e)p(e|f) \qquad (1)$$

In practice, we approximate this using N-best list $N(f)$ for the input f.

$$\operatorname*{argmin}_{e' \in N(f)} \sum_{e \in N(f)} L(e'|e)p(e|f) \qquad (2)$$

Although MBR works effectively for re-ranking single system hypotheses, it is challenging for system combination because the estimated $p(e|f)$ from different systems cannot be reliably compared. One practical solution is to use uniform $p(e|f)$ but this does not achieve Bayes Risk. GMBR corrects by parameterizing the loss function as a linear combination of sub-components using parameter $\boldsymbol{\theta}$:

$$L(e'|e;\boldsymbol{\theta}) = \sum_{k=1}^{K} \theta_k L_k(e'|e) \qquad (3)$$

For example, suppose the desired loss function is "1.0−BLEU". Then the sub-components could be "1.0−precision(n-gram) $(1 \leq n \leq 4)$" and "brevity penalty".

Assuming uniform $p(e|f)$, the MBR decision rule can be denoted as:

$$\operatorname*{argmin}_{e' \in N(f)} \sum_{e \in N(f)} L(e'|e;\boldsymbol{\theta}) \frac{1}{|N(f)|}$$
$$= \operatorname*{argmin}_{e' \in N(f)} \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e'|e) \qquad (4)$$

To ensure that the uniform hypotheses space gives the same decision as the original loss in the true space $p(e|f)$, we use a small development set to tune the parameter $\boldsymbol{\theta}$ as follows. For any two hypotheses e_1, e_2, and a reference translation e_r (possibly not in $N(f)$) we first compute the true loss: $L(e_1|e_r)$ and $L(e_2|e_r)$. If $L(e_1|e_r) < L(e_2|e_r)$, then we would want $\boldsymbol{\theta}$ such that:

$$\sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_1|e) < \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_2|e) \qquad (5)$$

so that GMBR would select the hypothesis achieving lower loss. Conversely if e_2 is a better hypothesis, then we want opposite relation:

$$\sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_1|e) > \sum_{e \in N(f)} \sum_{k=1}^{K} \theta_k L_k(e_2|e) \qquad (6)$$

Thus, we directly compute the true loss using a development set and ensure that our GMBR decision rule minimizes this loss.

2.2 Implementation

We implement GMBR for SMT system combination as follows.

First we run SMT decoders to obtain N-best lists for all sentences in the development set, and extract all pairs of hypotheses where a difference exists in the true loss. Then we optimize $\boldsymbol{\theta}$ in a formulation similar to a Ranking SVM [9]. The pair-wise nature of Eqs. 5 and 6 makes the problem amendable to solutions in "learning to rank" literature [6]. In this shared task, we used RIBES+BLEU (EJ,JE) and BLEU (CE) as our objective functions, so that we want to choose the best translation hypotheses both in terms of local view (BLEU) and global view (RIBES). There is one regularization hyperparameter for the Ranking SVM, which we set by cross-validation.

The development set of each translation task consisted of 2,000 sentences; we divided it halves and used the first half for tuning SMT parameters by Minimum Error Rate Training (MERT) [12], and the other half for training the GMBR system combination.

3. ENGLISH-TO-JAPANESE TASK

3.1 Run Configurations

For the English-to-Japanese task, we submitted three runs: two system combination results and one single system result.

EJPRIMARY: System combination of three systems (EJPO+BIGLM, EJPO+WFST, EJFOREST)

EJLIMITED: System combination of three systems (EJPO, EJPO+WFST, EJFOREST)

EJPO+WFST: Single system (described below)

They were based on four individual systems: three pre-ordering-based systems and one forest-to-string system.

EJPO: Pre-ordering + baseline lexicalized reordering (distortion limit: 6)

EJPO+BIGLM: EJPO + language model trained on large-scale language resources (distortion limit: 6)

EJPO+WFST: Pre-ordering + monotone translation with weighted finite state transducers (WFSTs)

EJFOREST: Forest-to-string translation (U-Tokyo) [14]

All the systems used all supplied bitext (excluding the development set) for training their phrase tables and word 5-gram language models, and tuned by MERT with the development set `pat-dev-2006-2007.txt`. Word segmentation was done by Mecab[1] (version 0.98 with IPAdic) for Japanese and `stepp` (included in Enju parser) for English. The decoders were WFST decoder of SOLON speech recognizer [7] for EJPO+WFST and Moses[2] for EJPO, EJPO+BIGLM. We present the technologies used in three pre-ordering-based systems in the following.

3.2 English Pre-ordering: Head Finalization

English and Japanese is one of the most challenging language pairs for SMT, because of their distant word orderings. Reordering is a computationally hard problem that requires $n!$ reorderings in the worst case. Conventional studies on SMT have proposed various reordering models: distance-based phrase reordering [10], lexicalized reordering model [13], hierarchical phrase-based translation [1] and syntax-based translation [16]. Recent work in such a distant language pair focuses on *pre-ordering*, which reorders source

[1] http://mecab.sourceforge.net/

[2] http://sourceforge.net/projects/mosesdecoder/

language words in target language order prior to SMT decoding [15, 2]. This efficiently works for long distance reordering and increases whole translation performance.

We use a rule-based pre-ordering for English-to-Japanese translation called *head finalization* [8], based on English syntactic parsing by Enju HPSG parser [3]. Since Japanese is a head-final language, we can emulate the Japanese word ordering by applying a simple rule to English parse trees, which moves syntactic heads toward the end of their siblings. Several supplement rules are also applied to reflect the nature of Japanese:

(1) Syntactic heads for coordination are not reordered

(2) Plural nouns (with NNS part-of-speech tag) are rewritten with singular ones

(3) Determiners "a", "an", and "the" are eliminated

(4) Place the following *pseudo*-particles immediately after verb arguments[4]:

`_va0`: arg1 of the sentence head verb

`_va1`: arg1 of other verbs

`_va2`: arg2 of verbs

Figure 1 shows an example of English parse tree and the corresponding head finalization result.

3.3 Monotone Translation with WFSTs

By pre-ordering, English sentences are now pre-ordered into Japanese word order; so we try to translate these head-final English (HFE) sentences by *monotone* translation. The monotone translation problem can easily be implemented by WFSTs that transduce HFE phrases into the corresponding Japanese phrases.

The phrase transduction can be decomposed to four sub-processes:

(1) Segment source sentence into phrase sequences

(2) Translate source phrases into target phrases

(3) Segment target phrases into target word sequence

(4) Score target word sequence with the n-gram language model

In practice, we further decompose (4) into

(4a) 1-gram-based scoring

(4b) (n-gram - 1-gram)-based scoring

for efficient decoding with on-the-fly n-gram language model composition [7]. Finally, we used the composed WFST:

$$d\,(P \circ d\,(T \circ d\,(S \circ L_1))) \circ L_{n-1}$$

where P, T, S, L_1, and L_{n-1} represent (1), (2), (3), (4a), and (4b) respectively, $d()$ means determinization of a WFST and $\circ$ means composition of two WFSTs. The topmost composition with L_{n-1} was done on-the-fly in decoding, others were static compositions. This WFST framework can achieve very fast decoding speeds (~3x faster than Moses).

Table 1: Results on the development set (English-to-Japanese)

System	BLEU	RIBES
EJPRIMARY	0.3622	0.7768
EJLIMITED	0.3463	0.7708
1) EJPO+WFST	0.3330	0.7637
2) EJPO+BIGLM	0.3486	0.7696
3) EJPO	0.3429	0.7678
4) EJFOREST	0.2697	0.6922

Table 2: Automatic evaluation results in English-to-Japanese task *(† indicates our own evaluation results)***.**

System	BLEU	NIST	RIBES
System combination			
EJPRIMARY	**0.3948**	**8.7134**	**0.7813**
EJLIMITED	0.3784	8.5444	0.7777
Individual systems			
1) EJPO+WFST	0.3683	8.3854	0.7729
2) EJPO+BIGLM†	0.3881	8.5965	0.7782
3) EJPO†	0.3683	8.3854	0.7754
4) EJFOREST (U-Tokyo)	0.2799	7.2575	0.6861
Other teams and organizer baselines			
G05-1 (best runner-up)	0.3403	8.2467	0.6905
BASELINEHPBMT	0.3166	7.7954	0.7200
BASELINEPBMT	0.3190	7.8811	0.7068

3.4 Use of Large-scale Monolingual Resources for N-gram Language Model

In addition to English-Japanese bitext of 3 million sentences, we also used Japanese monolingual resources of about 300 million sentences for training our 5-gram language model. They were from Japanese patent applications in 1993-2005, also provided by the organizers.

3.5 Results

3.5.1 System Combination on the Development Set

Table 1 shows the system combination and individual system results on the second half of the development set [5]. The combined systems achieved better BLEU and RIBES results than each individual system included.

3.5.2 Formal Run Results

Tables 2 and 3 show the official automatic and subjective evaluation results. Our primary run (EJPRIMARY) achieved the best results in all metrics. Our contrastive runs with only bitexts (EJLIMITED and EJPO+WFST) were even better than the other runs. We emphasize that our primary run was better than rule-based systems even in subjective evaluation.

A major advantage of our systems was the head finalization in English; it remarkably increased SMT performance both in automatic and subjective evaluation. Our GMBR system combination further increased its performance by supplemental use of other system outputs.

[3] http://www-tsujii.is.s.u-tokyo.ac.jp/enju/index.html

[4] arg1 and arg2 are swapped for passive verbs.

[5] Note that these are *open* test results with the first half of the original development set as the training data for system combination.

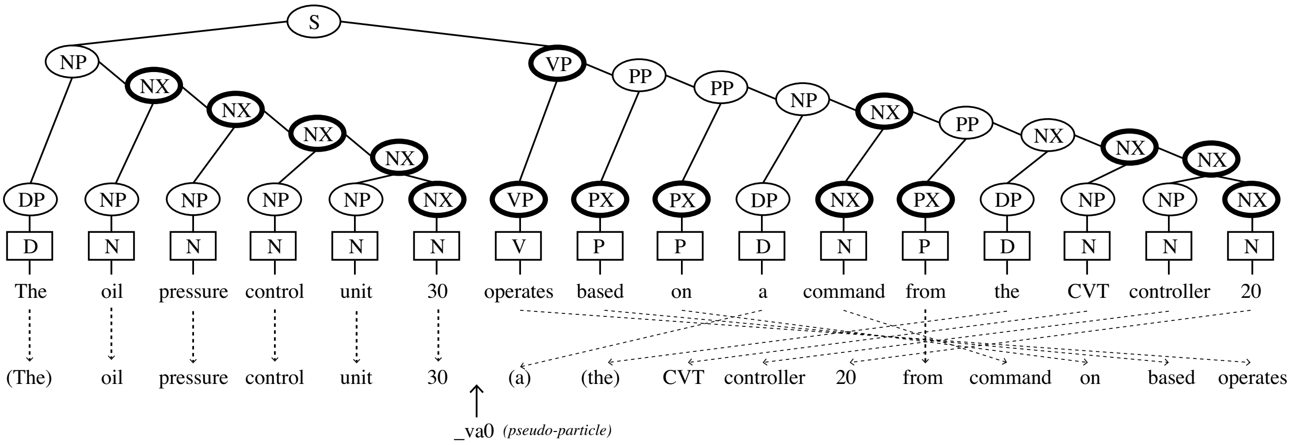
S
NP
NX
NX
NX
NX
VP
PP
PP
NP
NX
PP
NX
NX
NX
DP NP NP NP NP NX VP PX PX DP NX PX DP NP NP NX
D N N N N N V P P D N P D N N N
The oil pressure control unit 30 operates based on a command from the CVT controller 20
(The) oil pressure control unit 30 (a) (the) CVT controller 20 from command on based operates
_va0 (pseudo-particle)

Figure 1: An example English parse tree and its pre-ordering result.

Table 3: Official subjective evaluation results (English-to-Japanese).

System	Adequacy	Acceptability
EJprimary	**3.67**	**0.69**
EJpo+wfst	3.56	N/A
RBMT6-1	3.51	0.66
BaselineHPBMT	2.60	0.47
BaselinePBMT	2.48	0.46

4. JAPANESE-TO-ENGLISH TASK

4.1 Run Configurations

For the Japanese-to-English task, we submitted one system combination run:

JEprimary: System combination of three systems (JEbaseline, JEhier, JEpo)

This was based on three individual systems: baseline phrase-based and hierarchical phrase-based systems and a pre-ordering-based system.

JEbaseline: A baseline phrase-based system (distortion limit:12)

JEhier: A hierarchical phrase-based system (U-Tokyo) [14]

JEpo: Pre-ordering + monotone phrase-based system

All the systems used all supplied bitext (excluding the development set) for training their phrase tables and language models, and tuned by MERT with the development set `pat-dev-2006-2007.txt`. Word segmentation was done by Mecab (version 0.98 with IPAdic) for Japanese and `stepp` (included in Enju parser) for English. The decoders were Moses[6] for JEbaseline and JEpo.

4.2 Japanese Pre-ordering: POS-based Reordering of Modifiers

Pre-ordering in Japanese-to-English translation is not a trivial problem although pre-ordering in English-to-Japanese can be represented by a few rules. We developed a bit complex rules for reordering modifier chunks in Japanese.

The rules are based on our intuition that we can reorder Japanese chunks by their syntactic roles. We first applied *Cabocha* Japanese dependency parser[7] [11] to Japanese sentences, and then reorder Japanese modifier chunks with English-like order, according to "chunk precedence" below. The syntactic role of each chunk was determined by its function word (surface and part-of-speech (POS) tag) [8].

(1) Conjunction (`CONJ`)

(2) Subject (`SUBJ`)

(3) Verb (`VERB`)

(4) Direct object (`DOBJ`)

(5) Indirect object (`IOBJ`)

(6) Coodination (`COOD`)

(7) Others (`OTHER`)

For example, we generally move a verb chunk after its subject, a direct object after its verb, and so on. The rule sequence is shown in Algorithm 1.

4.3 Results

4.3.1 System Combination on the Development Set

Table 4 shows the system combination and individual system results on the second half of the development set. Similar to English-to-Japanese, the combined systems achieved better BLEU and RIBES results than each individual system included. Here, it is worth noting that the combination of the worst system (JEpo) increased the translation performance both in BLEU and RIBES.

[6] http://sourceforge.net/projects/mosesdecoder/

[7] http://code.google.com/p/cabocha/

[8] Cabocha annotates head and function words in chunks

Algorithm 1 Pseudocode for Japanese pre-ordering rules

```
for each modifier chunk $C_i$ do
  $h_i$ = headword($C_i$)
  $f_i$ = functionword($C_i$)
  $f_i^-$ = previous(functionword($C_i$)) # the word followed by the function word
  if POS($f_i$) == "particle"(助詞) then
    if POS_level2($f_i$) == "case marker"(格助詞) then
      if surface($f_i$) == $ga$(が) then
        $r_i$ = SUBJ
      else if surface($f_i$) == $o$(を) then
        $r_i$ = DOBJ
      else if surface($f_i$) == $ni$(に) | $e$(へ) | $to$(と) then
        $r_i$ = IOBJ
      else
        $r_i$ = OTHER
      else if POS_level2($f_i$) == "binding particle"(係助詞) then
        if surface($f_i$) == は (wa) then
          if POS_level2($f_i^-$) == "case particle"(格助詞) && surface($f_i^-$) == $ni$(に) | $e$(へ) | $to$(と) then
            $r_i$ = OTHER
          else
            $r_i$ = SUBJ
          end if
        else
          $r_i$ = OTHER
        end if
      end if
    else if POS_level2($f_i$) == "parallel marker"(並立助詞) then
      $r_i$ = COOD
    else
      $r_i$ = OTHER
    end if
  else if POS($f_i$) == "verb"(動詞) || POS($f_i$) == "auxiliary verb"(助動詞) then
    if POS($h_i$) == "verb"(動詞) && inflection($f_i$) == "conjunctive form"(連用形) then
      $r_i$ = VERB
    else if POS($h_i$) == "noun"(名詞) then
      $r_i$ = DOBJ
    else
      $r_i$ = OTHER
    end if
  else if POS($f_i$) == "conjunction"(接続詞) then
    $r_i$ = CONJ
  else
    $r_i$ = OTHER
  end if
end for
for each head chunk $H_i$ do
  reorderModifiers($H_i$) # reorder modifier chunks according to their precedence
end for
```

Table 4: Results on the development set (Japanese-to-English, case insensitive)

System	BLEU	RIBES
JEprimary	0.2939	0.7264
JEbaseline+JEhier	0.2865	0.7001
JEbaseine	0.2753	0.6832
JEpo	0.2629	0.6658
JEhier	0.2754	0.6946

Table 5: Automatic evaluation results in Japanese-to-English task *(† indicates our own evaluation results)*.

System	BLEU	NIST	RIBES
System combination			
JEprimary	0.2835	7.7934	0.7195
Individual systems			
JEbaseline†	0.2313	6.7785	0.6467
JEpo†	0.2778	7.4816	0.6700
JEhier (U-Tokyo)†	0.2605	7.5903	0.6732
Other teams and baselines			
G01-1 (best competitor)	0.3169	7.8161	0.7404
BaselineHPBMT	0.2895	7.7696	0.7064
BaselinePBMT	0.2861	7.7562	0.6758

4.3.2 Formal Run Results

Tables 5 shows the official automatic evaluation results. Our run (JEprimary) was slightly worse in BLEU but better in NIST and RIBES than the baseline systems. In the subjective evaluation, our run was the best among all SMT systems including the baselines, while the result itself was much worse than RBMT and HYBRID systems.

Our rule-based pre-ordering did not work in the individual system results, but it helped to increase system combination results. It is worth noting that we could increase translation performance from the baselines by combining other two systems which were a bit worse than the baselines in automatic evaluation metrics. Japanese-to-English pre-ordering was more difficult than English-to-Japanese; that may cause worse individual SMT system but it may generate *diverse* translations that are helpful for system combination.

4.3.3 Post-evaluation Results

The results above was degraded due to a serious mistake in testing; the test set Japanese sentences were tokenized by another tokenizer. We fixed the problem and conducted the experiments again. The post-evaluation results are shown in Table 7. The improvement from the original primary run was small but consistent among all automatic evaluation metrics.

Table 6: Official subjective evaluation results (Japanese-to-English).

System	Adequacy	Acceptability
JEprimary	2.75	0.49
G04-1 (RBMT)	3.67	0.71
RBMT1-1	3.53	0.67
G01-1 (HYBRID)	3.43	0.64
BaselineHPBMT	2.62	0.47
BaselinePBMT	2.44	0.45

Table 7: Post-evaluation results in Japanese-to-English task *(X^{post} indicate updated results after the evaluation period)*

System	BLEU	NIST	RIBES
System combination			
JEprimary^{post}†	0.2878	7.8583	0.7217
Individual system			
JEbaseline^{post}†	0.2675	7.6641	0.6816
JEpo†	0.2778	7.4816	0.6700
JEhier (U-Tokyo)†	0.2605	7.5903	0.6732

5. CHINESE-TO-ENGLISH TASK

5.1 Run Configurations

For the Chinese-to-English task, we submitted two system combination runs:

CEprimary: System combination of two systems (CEbaseline, CEhier)

CEexternal: System combination of two systems (CEadapt, CEhier)

They were based on three individual systems: baseline phrase-based and hierarchical phrase-based systems and one phrase-based system using additional, large-scale external bitexts.

CEbaseline: A baseline phrase-based system

CEhier: A hierarchical phrase-based system (U-Tokyo) [14]

CEadapt: A phrase-based system with adaptation using NIST CE data as additional resources

All the systems used all supplied bitext (excluding the development set) for training their phrase tables and language models, and tuned by MERT with the development set `ntc9-patentmt-develop-2k.txt`. Word segmentation was done by Stanford Chinese Segmenter[9] for Chinese, and `tokenizer.sed` for English. Both CEbaseline and CEadapt employed Moses as their SMT decoders.

5.2 Bayesian Word Alignment Adaptation

Chinese-to-English is a common SMT task and a lot of bitexts are now available. In order to improve our Chinese-to-English SMT, we utilized NIST OpenMT 2008 Chinese-to-English training data (4.8 million sentences with 107 million words, after cleaning) for word alignment adaptation [3]. Its key idea is, large-scale bitexts would help to improve word alignment between words in *general domain* even if the target domain (i.e. patent in this task) is a specific one.

5.3 Results

[9] http://nlp.stanford.edu/software/segmenter.shtml

Table 8: Results on the development set (Chinese-to-English, case insensitive)

System	BLEU
CEPRIMARY	0.3206
CEEXTERNAL	0.3200
CEBASELINE	0.2812
CEADAPT	0.2868
CEHIER	0.3108

Table 9: Automatic evaluation results in Chinese-to-English task *(† indicates our own evaluation results)*.

System	BLEU	NIST	RIBES
System combination			
CEPRIMARY	0.3026	8.0033	0.7647
CEEXTERNAL	0.3074	8.0031	0.7628
Individual systems			
CEBASELINE†	0.2735	7.3996	0.7398
CEADAPT†	0.2739	7.4489	0.7391
CEHIER (U-Tokyo)	0.3074	7.8917	0.7662
Other teams and baselines			
G1-1 (best competitor)	0.3944	8.9112	0.8327
BASELINEHPBMT	0.3072	7.9025	0.7719
BASELINEPBMT	0.2932	7.7498	0.7283

5.3.1 System Combination on the Development Set

Table 8 shows the system combination and individual system results on the second half of the development set [10]. Our system combination increased BLEU by 1.0 point from the best individual system (CEHIER).

5.3.2 Formal Run Results

Tables 9 and 10 show the official automatic and subjective evaluation results. The results of our two runs CEPRIMARY and CEADAPT were mixed compared to the baselines; our system combination did not improve the translation performance in the final test, in contrast to the development results above.

The reason for the drop in system combination performance from development to test results may be due to overtuning of GMBR parameters. In our post-evaluation, we found that GMBR's Ranking SVM in the Chinese-to-English task has relatively low regularization compared to other tasks. More regularization may have led to more stable results.

As for the domain adaptation performance, we saw small gains (0.5 points in BLEU) in development but no change in test. This may be because in-domain data is already relatively large, so that additional data does not have much impact.

6. CONCLUSIONS

In this shared task, we applied our GMBR system combination and achieved an improvement over the individual systems in all language pairs. We also included language dependent pre-ordering (English-from/to-Japanese) and Bayesian adaptation (Chinese-to-English) in our individual systems; their advantages were mixed among language pairs. We strongly emphasize our Head Finalization pre-ordering really helps in English-to-Japanese translation. On the other hand, our Japanese-to-English pre-ordering was not so good but it generated diverse translation hypotheses that helped to improve system combination results.

Table 10: Official subjective evaluation results (Chinese-to-English).

System	Adequacy	Acceptability
CEPRIMARY	3.23	n/a
G1-1	4.03	0.74
BASELINEHPBMT	3.29	0.48

7. ACKNOWLEDGMENTS

We would like to thank the PatentMT organizers for their great effort on this shared task. We also thank Dr. Takaaki Hori and Dr. Shinji Watanabe for their support for the use of SOLON WFSTs in our system, and Prof. Hideki Isozaki for valuable discussion on English-to-Japanese patent translation.

8. REFERENCES

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

[2] M. Collins, P. Koehn, and I. Kuĉerová. Clause restructuring for statistical machine translation. In *Proc. ACL*, pages 531–540, 2005.

[3] K. Duh, K. Sudoh, T. Iwata, and H. Tsukada. Bayesian adaptation of alignment matrices for machine translation. In *Proc. of MT Summit XIII*, 2011.

[4] K. Duh, K. Sudoh, X. Wu, H. Tsukada, and M. Nagata. Generalized bayes risk system combination for machine translation. In *Proc. of IJCNLP 2011*, 2011.

[5] I. Goto, B. Lu, K. P. Chow, E. Sumita, and B. K. Tsou. Overview of the patent machine translation task at the NTCIR-9 workshop. In *Proc. of NTCIR-9*, 2011.

[6] C. He, C. Wang, Y. X. Zhong, and R. F. Li. A survey on learning to rank. In *Proc. of International Conference on Machine Learning and Cybernetics*, 2008.

[7] T. Hori, C. Hori, Y. Minami, and A. Nakamura. Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 15:1352–1365, 2007.

[8] H. Isozaki, K. Sudoh, H. Tsukada, and K. Duh. Head finalization: A simple reordering rule for sov languages. In *Proc. of WMT 2010*, 2010.

[9] T. Joachims. Training linear SVMs in linear time. In *Proc. of KDD*, 2006.

[10] P. Koehn, F. J. Och, and D. Marcu. Statistical phrase-based translation. In *Proc. HLT-NAACL*, pages 263–270, 2003.

[11] T. Kudo and Y. Matsumoto. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002*, pages 63–69, 2002.

[10] Note again the system combination optimization was based only on BLEU, so we show only BLEU results.

[12] F. J. Och. Minimum error rate training in statistical machine translation. In *Proc. ACL*, pages 160–167, 2003.

[13] C. Tillmann. A unigram orientation model for statistical machine translation. In *Proc. HLT-NAACL*, pages 101–104, 2004.

[14] X. Wu, T. Matsuzaki, and J. Tsujii. Smt systems in the university of tokyo for NTCIR-9 patentmt. In *Proc. NTCIR-9*, 2011.

[15] F. Xia and M. McCord. Improving a statistical mt system with automatically learned rewrite patterns. In *Proc. COLING*, pages 508–514, 2004.

[16] K. Yamada and K. Knight. A syntax-based statistical translation model. In *Proc. ACL*, pages 523–530, 2001.