Nobody is Perfect:
ATR's Hybrid Approach to Spoken Language Translation

Michael Paul, Takao Doi, Youngsook Hwang, Kenji Imamura, Hideo Okuma, Eiichiro Sumita

ATR Spoken Language Communication Research Laboratories
Hikaridai 2-2-2, Keihanna Science City, 619-0288 Kyoto
{Michael.Paul,Takao.Doi,Youngsook.Hwang,Kenji.Imamura,Hideo.Okuma,Eiichiro.Sumita}@atr.jp

Abstract

This paper describes ATR's hybrid approach to spoken language translation and it's application to the IWSLT 2005 translation task. Multiple corpus-based translation engines are used to translate the same input, whereby the best translation among the element MT outputs is selected according to statistical models.

The evaluation results of the Japanese-to-English and Chinese-to-English translation tasks for different training data conditions showed the potential of the proposed hybrid approach and revealed new directions in how to improve the current system performance.

1. Introduction

Corpus-based approaches to machine translation (MT) have achieved much progress over the last decades. There are two main strategies used in corpus-based translation:

1. *Example-Based Machine Translation* (EBMT) [1]:
   EBMT uses the corpus directly. EBMT retrieves the translation examples that are best matched to an input expression and then adjusts the examples to obtain the translation.

2. *Statistical Machine Translation* (SMT) [2]:
   SMT learns statistical models for translation from corpora and dictionaries and then searches for the best translation at run-time according to the statistical models for language and translation.

Despite a high performance on average, these approaches can often produce translations with severe errors. However, different MT engines not always do the same error. Due to the particular output characteristics of each MT engine, quite different translation hypotheses are produced. Thus, combining multiple MT systems can boost the system performance by exploiting the strengths of each MT engine.

We propose a corpus-based approach that uses multiple translation engines in parallel. All engines translate the same input, whereby the best translation among the multiple MT outputs is selected according to multiple statistical language and translation models. The outline of our hybrid approach is given in Section 2.

The proposed system was applied to two translation directions (*Japanese-to-English*, *Chinese-to-English*) and three data tracks (*Supplied Data Track*, *Supplied+Tools Data Track*, *C-STAR Track*). The evaluation of the obtained results is given in Section 3.

2. System Description

We use an architecture in which multiple EBMT and SMT engines work in parallel and their outputs are passed to a post-process that selects the best candidate according to SMT models (cf. Figure 1).

This section is structured as follows: (1) eight corpus-based MT engines are introduced in Section 2.1; (2) the SMT-based approach to select the best translation out of multiple hypotheses is explained in Section 2.2; (3) the resources utilized for the IWSLT 2005 translation task are described in Section 2.3; and (4) a summary of the data tracks we participated in and an overview on which MT engines were utilized for the respective tracks is given in Section 2.4.

2.1. MT Engines

We employed the following four SMT and four EBMT systems: An SMT engine that uses an example-based decoding method [*SAT*] (cf. Section 2.1.1), an SMT engine that uses a phrase-based HMM translation model [*PBHMTM*] (cf. Section 2.1.2), a morpho-syntactically enriched phrase-based SMT engine [*MSEP*] (cf. Section 2.1.3), an SMT engine based on syntactic transfer [*HPATR2*] (cf. Section 2.1.4), an EBMT engine that incorporates word-level SMT methods [*HPATR*] (cf. Section 2.1.5), an EBMT engine based on hierarchical phrase alignments [*HPAT*] (cf. Section 2.1.6), an DP-match-driven EBMT engine [D^3] (cf. Section 2.1.7), and a translation memory system [*EM*] (cf. Section 2.1.8).

The translation knowledge of the eight MT systems is automatically acquired from a parallel corpus. The characteristics of the element MTs are summarized in Table 1.

2.1.1. SAT

SAT is an SMT system [3]. The decoder searches for an optimal translation by using SMT models starting from a decoder seed, i.e., the source language input paired with an initial translation hypothesis. In SAT, the search is initiated from

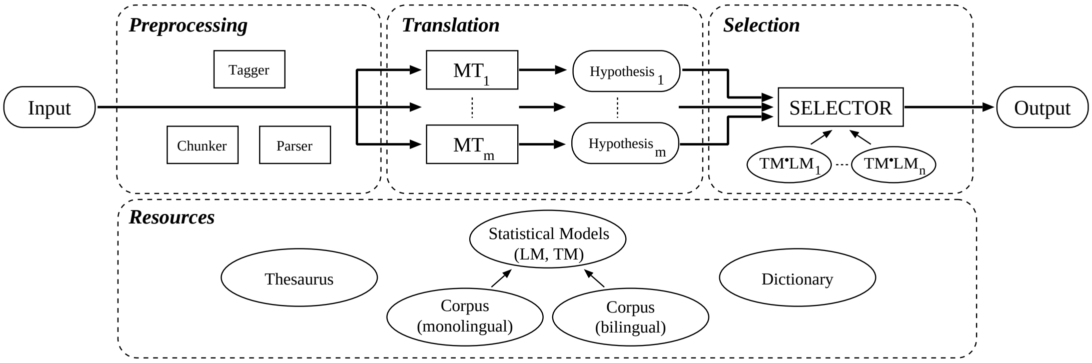

Figure 1: System outline

Table 1: Features of element MT engines

	SMT				EBMT			
	SAT	*PBHMTM*	*MSEP*	*HPATR2*	*HPATR*	*HPAT*	*D*^{3}	*EM*
Unit	sentence&word	phrase	phrase	phrase	phrase	phrase	sentence	sentence
Coverage	wide	wide	wide	wide	wide	wide	narrow	narrow
Quality	excellent	good	good	good	good	good	excellent	excellent
Speed	modest	slow	slow	modest	modest	fast	fast	fast
Resources	corpus	corpus	corpus, chunker	corpus, parser	corpus, parser	corpus, parser, thesaurus	corpus, thesaurus, bilingual dictionary	corpus

similar translation examples retrieved from a parallel corpus. The similarity measure used here is a combination of an edit-distance and tf/idf criteria as seen in the information retrieval framework [4]. The retrieved translations are modified by using a greedy search approach to find better translations [5].

2.1.2. PBHMTM

PBHMTM is a statistical MT system that is based on a phrase-based HMM translation model [6]. The model directly structures the phrase-based SMT approach in a Hidden Markov structure. The probability $P(\mathbf{f}|\mathbf{e})$ of translating a foreign source sentence f into a target language sentence e using noisy channel modeling is approximated by introducing two new hidden variables, $\bar{\mathbf{f}}$ and $\bar{\mathbf{e}}$, to explicitly capture the phrase translation relationship:

$$P(\mathbf{f}|\mathbf{e}) = \sum_{\bar{\mathbf{f}},\bar{\mathbf{e}}} \mathbf{P}(\mathbf{f}|\bar{\mathbf{f}},\bar{\mathbf{e}},\mathbf{e})\mathbf{P}(\bar{\mathbf{f}}|\bar{\mathbf{e}},\mathbf{e})\mathbf{P}(\bar{\mathbf{e}}|\mathbf{e}) \qquad (1)$$

The first term represents the probability that a phrase-segmented source language sentence $\bar{\mathbf{f}}$ can be reordered and generated as the source text of $\mathbf{f}$ (*Phrase Segmentation Model*). The second term indicates the translation probability of the two phrase sequences of $\bar{\mathbf{e}}$ and $\bar{\mathbf{f}}$ (*Phrase Translation Model*). The last term is the likelihood that the phrase-segmented target language sentence $\bar{\mathbf{e}}$ is generated from $\mathbf{e}$ (*Phrase Ngram Model*).

If the phrase segmented sentences $\bar{\mathbf{e}}$ and $\bar{\mathbf{f}}$ are expanded into corresponding lattice structures $\bar{\mathbf{E}}$ and $\bar{\mathbf{F}}$, then the approximation of the proposed models can be regarded as a Hidden Markov Model in which each source phrase in the lattice $\bar{\mathbf{F}}$ is treated as an observation emitted from a state, a target phrase, in the lattice $\bar{\mathbf{E}}$.

The decoder is a word-graph-based decoder [7], which allows the multi-pass decoding strategies to incorporate complicated submodel structures. The first pass of the decoding procedure generates the word-graph, or the lattice, of translations for an input sentence by using a beam search. On the first pass, the submodels of all phrase-based HMM translation models were integrated with the word-based trigram language model and the class 5-gram model. The second pass uses the A* strategy to search for the best path for translation on the generated word-graph.

2.1.3. MSEP

MSEP is a phrase-based SMT system that utilizes morpho-syntactic information such as part-of-speech and chunk information [8]. It exploits a phrase translation lexicon that is created using word-alignment results and chunk boundary information in a target language sentence. Reliable bilingual phrase pairs are identified using the statistical χ^2-test at a significance level α=0.05 over a given frequency threshold.

For selecting the most probable translation of a given source sentece, a log-linear model is used:

$$Pr_\Lambda(e_1^I|f_1^J) = \frac{exp(\sum_m \lambda_m h_m(e_1^I, f_1^J, a_1^J))}{\sum_{e_1^I,a_1^J} exp(\sum_m \lambda_m h_m(e_1^I, f_1^J, a_1^J))} \qquad (2)$$

where $h_m(e_1^I, f_1^J, a_1^J)$ is the *m-th feature* and λ_m is the weight of the feature.

In addition to the IBM model 4 features (word-based n-gram language model $Pr(e_1^I)$, lexicon model $t(f|e)$, fertility model $n(\phi|e)$, distortion probability d, and NULL translation model p_1), we incorporate the following features into the log-linear translation model:

- *Class-based n-gram model*: $Pr(e_1^I) = \prod_i \Pr(e_i|c_i) Pr(c_i|c_1^{i-1})$
- *Length model*: $Pr(l|e_1^I, f_1^J)$, whereby l is the length (number of words) of a translated target sentence.
- *Phrase matching score*: The translated target sentence is matched with phrase translation examples that are extracted from a parallel corpus based on bidirectional word alignment of phrase translation pairs. A score is derived based on the number of matches.

2.1.4. HPATR2

HPATR2 is a statistical MT system based on syntactic transfer [9]. The translation model of HPATR2 is defined as an inside probability of two parse trees, which is used to create probabilistic context-free grammar rules.

The system searches for the best translation that maximizes the product of the following probabilities, where $\mathcal{F}, \mathcal{E}$ are a source and a target parse trees, and θ, π are context-free grammar rules of the source and the target language, respectively.

- Probability of Source Tree Model
$$P(\mathcal{F}) = \prod_{\theta:\theta\in\mathcal{F}} P(\theta) \tag{3}$$
- Probability of Target Tree Model
$$P(\mathcal{E}) = \prod_{\pi:\pi\in\mathcal{E}} P(\pi) \tag{4}$$
- Probability of Tree-mapping Model
$$P(\mathcal{F}|\mathcal{E})P(\mathcal{E}|\mathcal{F}) = \prod_{\substack{\theta:\theta\in\mathcal{F},\\ \pi:\pi\in\mathcal{E}}} P(\theta|\pi)P(\pi|\theta) \tag{5}$$

A characteristic of HPATR2 is that not only word translations but also the translation of multi-word sequences is carried out by the syntactic transfer. Parsing hypotheses, which are multi-word sequences connected by context-free grammar rules, are created. The best hypothesis (parse tree and translation) is selected based on the above models.

Therefore, HPATR2 is an MT system that contains features of phrase-based SMT as well as syntax-based SMT.

2.1.5. HPATR

HPATR is an extension of the example-based HPAT system (cf. Section 2.1.6) that incorporates a word-based statistical MT system [10]. Similar to HPAT, an EBMT module based on syntactic transfer is used to generate translation candidates that have minimum semantic distances. However, word selection is not performed during transfer, but all possible word translation candidates are generated.

In a second step, an SMT module using a lexicon model and an n-gram language model is exploited to search for the best translation that maximizes the product of the probabilities. Therefore, HPATR selects the best translation among the output of example-based MT using models of statistical MT from the viewpoints of adequacy of word translation and fluency of the target sentence.

2.1.6. HPAT

HPAT is an example-based MT system based on syntactic transfer [11]. The most important knowledge in HPAT are transfer rules, which define the correspondences between source and target patterns. The transfer rules can be regarded as synchronized context-free grammar rules.

When the system translates an input sentence, the sentence is first parsed by using the source side of the transfer rules. Next, a tree structure of the target language is generated by mapping the source grammar rules to the corresponding target rules. When non-terminal symbols remain in the target tree, target words are inserted by referring to a translation dictionary.

Ambiguities, which occur during parsing or mapping, are resolved by selecting the rules that minimize the semantic distance between the input words and source examples of the transfer rules. In general, the automatic acquisition process generates many redundant rules. To avoid this problem, HPAT optimizes the transfer rules by removing redundant rules (*feedback cleaning*, [12]) in order to increase an automatic evaluation score.

2.1.7. D^3

D^3 (*DP-match Driven transDucer*) is an EBMT system that exploits DP-matching between word sequences [13]. In the translation process, D^3 retrieves the most similar source sentence from a parallel corpus for an input sentence.

The similarity is calculated based on the counts of insertion, deletion, and substitution operations, where the total is normalized by the sum of the lengths of the word sequences. Substitution considers the semantic distance between two substituted words and is defined as the division of K, the level of the least common abstraction in the thesaurus of two words, by N, the height of the thesaurus [14].

According to the difference between the input sentence and the retrieved source sentence, the translation of the retrieved source sentence is modified by using dictionaries.

2.1.8. EM

EM is a translation memory system that matches a given source sentence against the source language parts of translation examples extracted from a parallel corpus. In case an exact match can be achieved, the corresponding target language sentence will be used. Otherwise, the system fails to output a translation.

2.2. Selection of the Best MT Engine Output

We use an SMT-based method of automatically selecting the best translation among outputs generated by multiple MT systems [15]. This approach scores MT outputs by using multiple language (LM) and translation model (TM) pairs trained on different subsets of the training data. It uses a statistical test to check whether the average TM·LM score of one MT output is significantly higher than those of another MT output. The SELECTOR algorithm is summarized in Figure 2.

```
proc SELECTOR( $Input, Corpus, n, MT_1, \ldots, MT_m$ ) ;
begin
    (* initalize statistical models *)
    for each $i$ in $\{1, \ldots, n\}$ do
        $Corpus_i \leftarrow$ subset($Corpus$) ;
        $TM_i \leftarrow$ translation-model($Corpus_i$) ;
        $LM_i \leftarrow$ language-model($Corpus_i$) ;
    od ;
    (* score MT outputs using multiple TM·LM pairs *)
    $HypScores \leftarrow \{\}$ ;
    for each $MT$ in $\{MT_1, \ldots, MT_m\}$ do
        $Scores \leftarrow \{\}$ ;
        $Hypothesis \leftarrow$ translate($Input, MT$) ;
        if $Hypothesis$ then
            (* assign multiple TM·LM scores to each hypothesis *)
            for each $i$ in $\{1, \ldots, n\}$ do
                $Scores \leftarrow Scores \cup \{\text{TM}_i \cdot \text{LM}_i(Hypothesis)\}$ ;
            od ;
            $HypScores \leftarrow HypScores \cup \{Hypothesis, Scores\}$ ;
        fi ;
    od ;
    (* multiple comparison test based on Kruskal-Wallis test *)
    $i_{sel} \leftarrow 1$ ;
    for each $i$ in $\{2, \ldots, m\}$ do
        $i_{better} \leftarrow$ test($HypScores(i_{sel}), HypScores(i)$) ;
        if $i_{better}$ then
            $i_{sel} \leftarrow i_{better}$ ;
        fi ;
    od ;
    return( $HypScores(i_{sel})$ ) ;
end ;
```

Figure 2: SELECTOR algorithm

In order to detect a significant difference, the proposed method first prepares multiple subsets of the full parallel text corpus according to n-fold[1] cross validation [16] and trains pairs of language and translation models on the resepective subsets.

The algorithm assumes a priority order in the given MT engines, i.e., the indices i of MT_i should indicate an increasing level (i=1 - high, …, i=n - low) of translation quality of the utilized MT systems. Such a priority order can be obtained by evaluating the respective system performance using a separate development set[2].

Given an input sentence, m translation hypotheses are produced by the element MT engines, whereby n different statistical scores (TM·LM$_i$) are assigned to each hypothesis. In order to check whether the highest score is significantly different from the others, a multiple comparison test based on the Kruskal-Wallis test [17] is used. If the MT output is significantly better, this output is selected. Otherwise, the output of the best performing MT engine according to the given priority order is selected.

The performance of the selection method depends on the predefined priority order of the MT engines. For the experiments described in Section 3, the respective priority order for each data track were determined as follows:

1. The highest priority was given to the *EM* engine.
2. The rest of the MT engines were sorted according to the WER scores obtained for the development set.
3. For each subset {MT$_1$,…,MT$_j$} (1≤j≤n-1) of top-scoring j MT engines, all order combinations were used to translate the development set. The combination achieving the lowest WER score was chosen for the translation of the official test set.

2.3. Resources

The participants were supplied with 20,000 sentence pairs for each translation direction that are subsets of the BTEC corpus introduced in Section 2.3.1. For the Japanese and Chinese parts, word segmentations were provided based on the output of the speech recognition engine utilized in this years' workshop. In addition, two development sets of additional 506 and 500 sentences, respectively, including up to 16 reference translations, were provided to the participants.

2.3.1. *Basic Travel Expressions Corpus*

The *Basic Travel Expressions Corpus* (BTEC) is a collection of sentences that bilingual travel experts consider useful for people going to or coming from another country and cover utterances for every potential subject in travel situations [18]. The original Japanese-English corpus has been translated into several languages by members of the Consortium for Speech Translation Advanced Research (C-STAR)[3] resulting in a multilingual spoken language corpus consisting of 172K sentences/language.

Moreover, we used additional training data from the same domain for the Japanese-to-English translation task. In total, 541K of aligned sentence pairs were available for training purposes. Besides the data sets of the *Supplied Data Track* (cf. Section 2.4), all training data were preprocessed using in-house morphological analysis tools. Details of the utilized corpus are given in Table 2, where *word token* refers to the number of words in the corpus and *word type* refers to the vocabulary size.

2.3.2. *NLP Tools and Dictionaries*

Besides the parallel text corpus, we also used in-house NLP tools (tagger, chunker, parser) for the preprocessing of the

[1] For this years' workshop, we randomly divided the training corpus into three subsets ($Corpus_i$; 1≤i≤3) and trained three different translation and language model pairs on all pairwise combinations of the subsets ($Corpus_1 \cup Corpus_2$, $Corpus_1 \cup Corpus_3$, $Corpus_2 \cup Corpus_3$).

[2] We used a development set of the supplied corpus and sorted the MT engines according to an automatic evaluation score (*WER*, cf. Section 3).

[3] http://www.c-star.org/

Table 2: Training corpus statistics

data track	language	sentence count total	sentence count unique	avg. length	word tokens	word types
C	Japanese	541,665	382,446	9.6	5,564,159	40,064
	Chinese	172,170	87,845	6.7	1,158,039	17,570
	English	541,665	343,764	8.3	4,520,340	21,737
T	Japanese	20,000	19,078	9.8	196,513	8,376
	Chinese	20,000	18,668	8.1	162,808	9,040
	English	20,000	19,914	9.2	183,995	5,464
S	Japanese	20,000	18,906	8.6	171,259	9,251
	Chinese	20,000	19,267	8.8	176,174	8,683
	English	20,000	19,902	7.7	154,483	6,937

training data and additional knowledge resources like bilingual dictionaries and a thesaurus [19] as summarized in Table 1.

2.4. Track Participation

We took part in the following translation tasks using the manual transcriptions as the input of the MT engines.

Translation Direction:	(**JE**)	Japanese-to-English
	(**CE**)	Chinese-to-English
Data Track:	(**C**)	C-STAR Track
	(**T**)	Supplied+Tool Data Track
	(**S**)	Supplied Data Track

Each MT engine was trained on the resources permitted for the respective data track. In the case of the *Supplied Data Track*, the training data was limited to the supplied corpus only, i.e., preprocessing tools like tagger, chunker, parser and external dictionaries were not allowed. Therefore, only three element MT engines (*PBHMTM*, *SAT*, *EM*) were used for the *Supplied Data Track*. Moreover, the HPAT engine for CE could not be prepared in time for the workshop.

In addition, the SELECTOR engine was tuned to obtain the best WER performance as described in Section 2.2. Therefore, not all MT engines were used for the official run submissions. Table 3 gives an overview of the used MT engines for each data track, whereby the respective priority orders are given in Table 4.

3. Evaluation

For the automatic evaluation of the MT outputs, we used the online evaluation server[4], whereby the following scoring metrics were used:

- *BLEU*: the geometric mean of n-gram precision for the translation results found in reference translations.
- *NIST*: a variety of BLEU using the arithmetic mean of weighted n-gram precision.
- *METEOR* which scores unigram matches using different criteria (exact matching, stemmed matching, and synonym matching).

Table 3: MT engines used for official run submissions

MT engine	JE C	JE T	JE S	CE C	CE T	CE S
SAT	○	○	○	○	○	○
PBHMTM	○	○	○	○	○	○
MSEP	×	○	N/A	○	○	N/A
HPATR2	○	○	N/A	○	○	N/A
HPAT	×	○	N/A	N/A	N/A	N/A
HPATR	×	×	N/A	○	○	N/A
D3	○	○	N/A	○	○	N/A
EM	○	○	○	○	○	○
Σ	5	7	3	7	7	3

○ = used × = not used for official runs N/A = output not available

Table 4: Priority order of MT engines

language	data track	priority order
JE	C	EM>D3>HPATR2>SAT>PBHMTM
	T	EM>D3>HPAT>HPATR2>PBHMTM>SAT>MSEP
	S	EM>PBHMTM>SAT
CE	C	EM>D3>HPATR2>HPATR>MSEP>PBHMTM>SAT
	T	EM>MSEP>D3>HPATR>PBHMTM>HPATR2>SAT
	S	EM>PBHMTM>SAT

- *Word Error Rate* (WER) which penalizes edit operations for the translation output against reference translations.
- *Position independent WER* (PER) which penalizes without considering positional disfluencies.
- *GTM* which measures the similarity between texts by using a unigram-based F-measure.

In contrast to WER/PER, higher BLEU/NIST/METEOR/GTM scores indicate better translations. Besides for GTM (one reference translation), up to 16 human reference translations were used for the automatic scoring.

In addition, the translation quality was judged by an English native speaker based on the *fluency* and *adequacy* criteria, whereby *fluency* refers to the degree to which the translation is well-formed according to the grammar of the target language and *adequacy* refers to the degree to which the translation communicates the information present in the reference output. The evaluator assigns a fluency score in the range from 5 (=*flawless*) to 1 (=*incomprehensible*) and an adequacy score in the range from 5 (=*all information*) to 1 (=*none of it*) to each translation. The system scores for fluency and adequacy are obtained as the average of all sentence scores.

3.1. Official Run Submissions

The evaluation results of the official run submissions for the IWSLT 2005 translation task are summarized in Table 5. The

[4] http://penance.is.cs.cmu.edu/iwslt2005/eval_servers.html

Table 5: Evaluation results of offical run submissions

lang	data	Automatic Evaluation						Subjective Evaluation†	
uage	track	BLEU	NIST	METEOR	WER	PER	GTM	Fluency	Adequacy
JE	C	**0.6873**	**10.7375**	**0.8102**	**0.2768**	**0.2286**	**0.6934**	**4.52**	**4.27**
	T	0.4774	8.1720	0.6658	0.4349	0.3742	0.5520	3.68	3.33
	S	0.3744	7.7368	0.6008	0.5568	0.4570	0.4822	3.23	2.68
CE	C	**0.5031**	**8.6875**	**0.6845**	**0.4389**	**0.3727**	**0.5898**	**4.14**	**3.26**
	T	0.3804	6.7540	0.5819	0.5439	0.4624	0.4950	3.61	2.57
	S	0.3938	8.0004	0.6291	0.5235	0.4276	0.5533	3.54	2.73

† This subjective evaluation is carried-out in-house according to the evaluation specifications of IWSLT.

best performing system for each of the translation directions is marked with bold-face.

3.2. Effects of Training Data Size

At submission time, 541K sentence pairs of the same domain as the supplied corpus were available for JE, but only 172K sentence pairs for CE. These were used for the official run submissions of the C-STAR track (cf. Table 5). Afterwards the CE corpus was extended to 541K and additional runs for CE using 541K sentence pairs and JE using 172K sentence pairs were evaluated using the online evaluation server. Figure 3 illustrates how the system performance improves for larger amounts of training data.

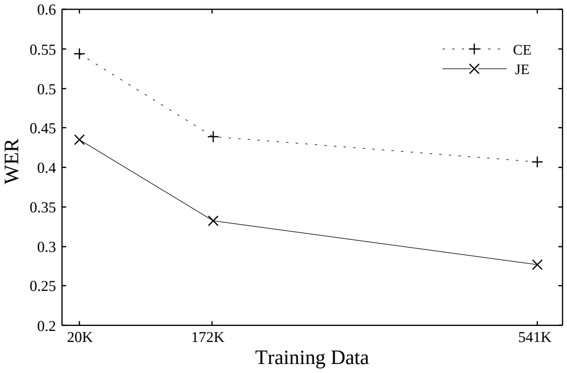

Figure 3: Effects of variable amounts of training data

In order to illustrate the similarity between the training data used for the respective data tracks and the test data, the language perplexity[5] (PPL) of the source language input and target language translations is summarized in Table 6. In addition, the average sentence length in words, the total word counts and the percentage of out-of-vocabulary (OOV) words of our official run submissions are given.

The results show different tendencies for CE and JE. An analysis of the data sets revealed, that the increase in language perplexity for the CE data tracks is due to Chinese text style differences between the utilized training corpus and the IWSLT05 test set.

Table 6: Run Submission Statistics

lang	data	source		target			
uage	track	PPL	OOV	PPL	OOV	words	length
JE	C	13.2	0.3	14.3	0.1	3006	5.9
	T	19.5	1.7	23.0	0.3	2925	5.7
	S	40.6	7.1	50.7	15.8	3139	6.2
CE	C	165.4	1.8	17.8	0.1	3050	6.0
	T	80.5	3.8	16.6	0.4	2979	5.8
	S	52.8	3.8	39.5	15.6	3141	6.2

3.3. Effects of NLP tools

For JE, a large difference in system performance can be seen for the *Supplied Data Track* and the *Supplied+Tools Data Track*. This gain is partially due to the usage of morphological analyzing tools that normalize variations of surface words, e.g., usage of *hiragana* or *kanji* in Japanese. In order to investigate the effects of prepocessing tools, we compared the *Supplied Data Track* submission (selection of three element MT outputs) with the SELECTOR output when applied to the respective three MT engines (3MT = SAT, PBHMTM, EM) of the *Supplied+Tool Data Track*. The evaluation results summarized in Table 7 show that a medium improvement of 3.5% in WER is achieved for JE. However, word segmentation differences and the lower coverage of our inhouse Chinese tagging tool resulted in a degradation in performance for the CE task.

Table 7: Effects of tagging tools

language	data track	WER
JE	T (3MT)	**0.5221**
	S	0.5568
CE	T (3MT)	0.5913
	S	**0.5235**

The effects of other NLP tools, like chunker and parser, cannot be compared directly to the results obtained for the *Supplied Data Track*. However, they are important in our hybrid approach, because they cover morpho-syntactic information that cannot be obtained from plain text and thus lead to the generation of translation hypotheses with quite different characteristics.

[5] We used the SRI Language Modeling Toolkit (version 1.4.4), which is available at http://www.speech.sri.com/projects/srilm/download.html.

Table 8: Evaluation of element MT engines (WER)

MT engine	JE			CE		
	C	T	S	C	T	S
SAT	0.3404	0.5541	0.5664	0.5186	0.6590	0.5690
PBHMTM	**0.3268**	**0.5310**	**0.5589**	0.5180	0.6071	**0.5366**
MSEP	0.3956	0.5384	N/A	**0.4785**	**0.5645**	N/A
HPATR2	0.3457	0.5478	N/A	0.5685	0.6436	N/A
HPAT	0.4526	0.5427	N/A	N/A	N/A	N/A
HPATR	0.4137	0.5507	N/A	0.7148	0.6890	N/A
D3	0.3971	0.5650	N/A	0.6179	0.7137	N/A
EM	0.5995	0.8949	0.9426	0.9413	0.9775	0.9659

N/A = output not available

Table 9: Element MT engines of selected hypotheses (%)

MT engine	JE			CE		
	C	T	S	C	T	S
SAT	9.9	3.0	5.9	5.3	1.8	19.2
PBHMTM	16.4	**23.3**	**84.4**	8.9	13.7	**76.1**
MSEP	×	10.9	N/A	**39.1**	**68.8**	N/A
HPATR2	17.2	12.0	N/A	23.3	3.3	N/A
HPAT	×	17.4	N/A	N/A	N/A	N/A
HPATR	×	×	N/A	4.7	6.1	N/A
D3	12.1	19.4	N/A	11.1	2.8	N/A
EM	**44.4**	14.0	9.7	7.6	3.5	4.7

× = not used for official runs N/A = output not available

3.4. Effects of Multi-Engine Approach

In order to investigate the effects of our hybrid approch, we compared the SELECTOR output towards the system performance of each element MT engine (cf. Table 8).

The results of the element MT engines show large variations in the WER scores, whereby the SMT engines performed best. However, this does not mean that translation hypotheses produced by EBMT engines are not useful at all. Table 9 gives the percentage of translation hypotheses of each MT engine selected for the official run submissions.

Comparing the results of the best performing MT engines (marked in bold-face in Table 8) for all data tracks towards the respective SELECTOR outputs reveales that the selection algorithm outperforms all element MT engines (cf. Table 10) gaining 4-5% in WER for the *C-STAR Track* submissions and even up to 10% in WER for the *Supplied+Tools Data Track*.

In order to investigate in an upper boundary for the proposed selection algorithm, we performed and "oracle" translation experiment. Each input sentence was translated by all element MT engines and the translation hypothesis with the lowest WER (compared to the reference translations) was output as the translation, i.e., the ORACLE system simulates an optimal selection method for the given element MT engines based on the WER criterion.

The comparison of the achieved gains[6] of the SELECTOR and the ORACLE method given in Table 10 reveals that despite a significant improvement in the overall system performance, the proposed SMT-based selection method still underachieved its task.

Table 10: Effects of SELECTOR approach (WER)

language	data track	best MT	SELECTOR (gain)	ORACLE (gain)
JE	C	0.3268	0.2768 (– **0.0500**)	0.1696 (– **0.1572**)
	T	0.5310	0.4349 (– **0.0961**)	0.3006 (– **0.2304**)
	S	0.5589	0.5568 (– **0.0021**)	0.4600 (– **0.0989**)
CE	C	0.4785	0.4389 (– **0.0396**)	0.3246 (– **0.1539**)
	T	0.5645	0.5439 (– **0.0206**)	0.3996 (– **0.1649**)
	S	0.5366	0.5235 (– **0.0103**)	0.4526 (– **0.0840**)

4. Conclusion

In this paper, a hybrid corpus-based approach to spoken language translation was evaluated on the IWSLT 2005 translation task for Japanese-to-English and Chinese-to-English. Each input sentence was translated using up to eight MT engines, whereby the best translation was selected based on statistical models. High performances were achieved for both translation directions. The analysis of the evaluation results revealed, that:

- an increase in training data leads to improved results
- the preprocessing of the training data is important to achieve high translation quality
- the translation quality of the element MT engines ranged from *medium* to *high*.
- the proposed selection method outperformed all element MT engines gaining 4-5% in WER towards the best performing MT engine.
- despite a significant improvement in the overall system performance, the SMT-based selection method underachieved its task. An offline evaluation of the translation results showed that an improvement of up to 16% in WER towards the best performing MT engine could be possible for the IWSLT 2005 translation task.

Future research will have to incorporate additional features besides statistical model scores in the selection process in order to tap the full potential of the element MT engines.

5. Acknowledgements

The research reported here was supported in part by a contract with the National Institute of Information and Communications Technology entitled *"A study of speech dialogue translation technology based on a large corpus"*. We are especially thankful to Yasuhiro Akiba and Taro Watanabe for their contributions to our efforts. In addition, the authors

[6] The WER figures in parantheses represent the gain of the respective selection method towards the best performing element MT.

would like to thank Kadogawa-Shoten for providing us with the Ruigo-Shin-Jiten thesaurus.

6. References

[1] M. Nagao, “A framework of a mechanical translation between japanese and english by analogy principle,” in *Proc. of the International NATO Symposium on Artificial and Human Intelligence*. North Holland: Elsevier, 1984, pp. 173–180.

[2] P. Brown, S. D. Pietra, V. D. Pietra, and R. Mercer, “A statistical approach to machine translation,” *Computational Linguistics*, vol. 16(2), pp. 79–85, 1990.

[3] T. Watanabe and E. Sumita, “Example-based decoding for statistical machine translation,” in *Proc. of MT Summit IX*, New Orleans, USA, 2003, pp. 410–417.

[4] C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press, 1999.

[5] U. Germann, M. Jahr, K. Knight, D. Marcu, and K. Yamada, “Fast decoding and optimal decoding for machine translation,” in *Proc. of ACL*, Toulouse, France, 2001.

[6] E. Sumita, Y. Akiba, T. Doi, A. Finch, K. Imamura, H. Okuma, M. Paul, M. Shimohata, and T. Watanabe, “EBMT, SMT, Hybrid and More: ATR Spoken Language Translation System,” in *Proc. of International Workshop on Spoken Language Translation*, Kyoto, Japan, 2004, pp. 13–20.

[7] H. N. Nicola Ueffing, Franz Josef Och, “Generation of word graphs in statistical machine translation,” in *Proc. of EMNLP*, Philadelphia, USA, 2002, pp. 156–163.

[8] Y. Hwang, H. Chung, and H. Rim, “Weighted probabilistic sum model based on decision tree decomposition for text chunking,” *IJCPOL*, vol. 16(1), pp. 1–20, 2003.

[9] K. Imamura, H. Okuma, and E. Sumita, “Practical approach to syntax-based statistical machine translation,” in *Proc. of Machine Translation Summit X*, Phuket, Thailand, 2005, p. (to appear).

[10] K. Imamura, H. Okuma, T. Watanabe, and E. Sumita, “Example-based machine translation based on syntactic transfer with statistical models,” in *Proc. of COLING*, Geneva, Switzerland, 2004, pp. 99–105.

[11] K. Imamura, “Application of translation knowledge acquired by hierarchical phrase alignment for pattern-based MT,” in *Proc. of TMI*, Keihanna, Japan, 2002, pp. 74–84.

[12] K. Imamura, E. Sumita, and Y. Matsumoto, “Feedback cleaning of machine translation rules using automatic evaluation,” in *Proc. of ACL*, Sapporo, Japan, 2003, pp. 447–454.

[13] E. Sumita, “Example-based machine translation using DP-matching between word sequences,” in *Proc. of ACL, Workshop: Data-Driven Methods in Machine Translation*, Toulouse, France, 2001, pp. 1–8.

[14] E. Sumita and H. Iida, “Experiments and prospects of example-based machine translation,” in *Proc. of ACL*, 1991, pp. 185–192.

[15] Y. Akiba, T. Watanabe, and E. Sumita, “Using language and translation models to select the best among outputs from multiple mt systems,” in *Proc. of COLING*, Taipei, Taiwan, 2002, pp. 8–14.

[16] T. Mitchell, *Machine Learning*. New York, USA: The McGraw-Hill Companies Inc., 1997.

[17] Y. Hochberg and A. Tamhane, *Multiple Comparison Procedures*. New York, USA: Wiley, 1987.

[18] G. Kikui, E. Sumita, T. Takezawa, and S. Yamamoto, “Creating corpora for speech-to-speech translation,” in *Proc. of EUROSPEECH03*, Geneve, Switzerland, 2003, pp. 381–384.

[19] S. Ohno and M. Hamanishi, *Ruigo-Shin-Jiten*. Kadokawa, 1984.