The NiCT-ATR Statistical Machine Translation System for the IWSLT 2006 Evaluation

Ruiqiang Zhang, Hirofumi Yamamoto, Michael Paul, Hideo Okuma, Keiji Yasuda,
Yves Lepage, Etienne Denoual, Daichi Mochihashi, Andrew Finch,Eiichiro Sumita

National Institute of Information and Communications Technology
ATR Spoken Language Communication Research Laboratories
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0288, Japan
firstname.lastname@{atr, nict.go}.jp

Abstract

This paper describes the NiCT-ATR statistical machine translation (SMT) system used for the IWSLT 2006 evaluation compaign. We participated in all four language pair translation tasks (CE, JE, AE and IE) and all two tracks (OPEN and CSTAR). We used a phrase-based SMT in the OPEN track and a hybrid multiple translation engine in the CSTAR track. We also equipped our system with some of new preprocessing and post-processing techniques for Chinese word segmentation, named entity translation, punctuation and capitalization, sentence splitting, and language model adaptation. Our experiments show these features significantly improved our system.

1. Introduction

Phrase-based statistical machine translation (SMT) has progressed over the years and is the primary approach for SMT research. This approach is used by 80% of the systems participating in the NIST 2006 machine translation evaluation. Our main translation engine for this year's IWSLT evaluation, TATR, is also a phrase-based SMT.

The hybrid multiple engine approach, that was used last year [1], was used again this year. But we replaced the 2005 SMTs (PBHMTM and SAT) with TATR, partly for simplification reasons. In addition to TATR, two other engines are included in this year's hybrid system: HPATR3, a SMT based on syntactic transfer; and EM, the translation memory based on exact match.

We employed new approaches for pre-processing, post-processing, and language modeling. We used subword-based Chinese word segmentation [2]. This word segmentation achieved the highest F-score rate for the second Sighan test data, and can recognize numerical expressions and foreign names. We built a conversion model to implement capitalization and punctuation by using the maximum entropy principle and the conditional random field (CRF) approach, which can integrate long-range features to enhance performance. We applied sentence-splitting techniques to all languages. This approach significantly improved CE and JE translation. For language modeling, we used a new language model adaptation approach that can divide training data by topic. For each topic, a topic-dependent language model was built and applied to input belonging to this topic at the time of translation. We found this approach also improved translations.

In this year's evaluation, we participated in all four language pair translation tasks and two tracks: OPEN and CSTAR track. A list of all tests is shown in Table 1, where "√" indicates we participated in the test and "×" means we did not. We participated in 14 out of all the 18 tests.

Our translation system flow chart is illustrated in Fig. 1. Before the input is translated by the MT engines, it is preprocessed with a series of preprocessing methods: word segmentation and sentence-splitting. We used three translation engines, TATR, HPATR3 and EM, and used *Selector* for the CSTAR track. But we used only one translation engine, TATR, for the OPEN track. The final output is generated after the post-processing module for the punctuation and capitalization.

In the following sections, section 2 describes our word segmentation methods. Section 3 describes our language model adaptation. Section 4 describes the subword-based name entity translation for Chinese. Section 5 describes the translation engines, TATR and *Selector*. Section 6 describes our CRF-based punctuation and capitalization. Section 7 presents our evaluation results. Section 8 provides our conclusions and comments.

2. Preprocessing

2.1. Arabic segmentation

Of the released data, we threw away all end-of-utterance markers. However, we kept any sentence markers that were in the middle of an utterance, but standardized them to the same unique marker. This punctuation preprocessing was performed on the Arabic data as well as on the English data.

The morphological analysis of Arabic was performed using the BAMA as released by the LDC consortium [3]. This implied that we had first to convert the encoding from UTF-8 into the Windows-1256 encoding.

Table 1: List of IWSLT 2006 tests that we have participated in

language	CE			JE		AE		IE	
track	spontaneous	read	correct	read	correct	read	correct	read	correct
OPEN	√	√	√	√	√	√	√	√	√
CSTAR	√	√	√	√	√	×	×	×	×

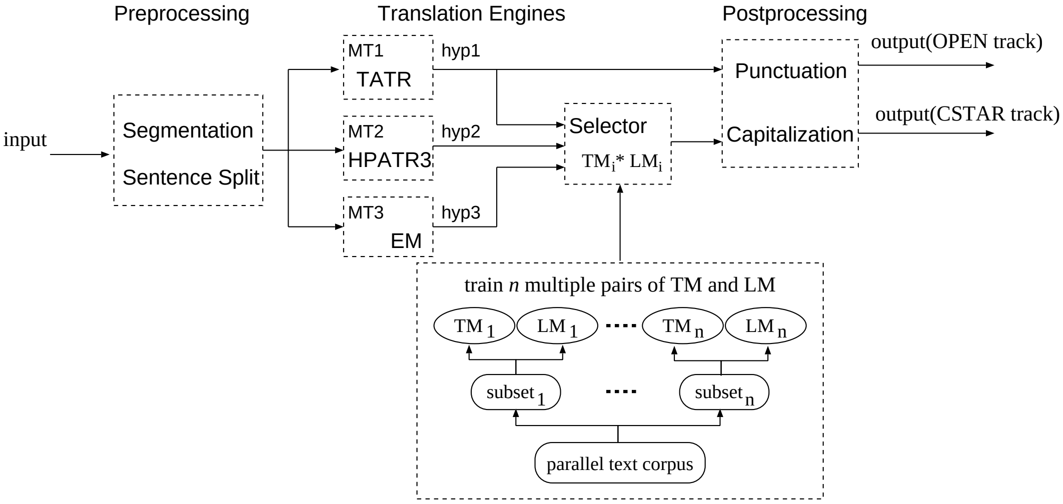

Figure 1: Overview of our translation system

The Arabic writing system composes the base of a word with a set of proclitics and enclitics: conjunctions, particles, the article and pronouns, etc. Such agglutinated forms are highly ambiguous, and one such inflected form yields on average about 7 different readings.

As the size of the supplied data is small, we followed the conclusions of [4] and opted for a morphological analysis of Arabic. The output of the BAMA is a list of readings of agglutinated forms in their transcribed form. In contrast, our approach differs from the technique used in [5] where the MADA tool is used to select the best hypothesis among the candidate parses, we did not perform any disambiguation but chose to select, in a consistent way, the first hypothesis of the BAMA output. The input to the machine translation system was the *Buckwalter* transcribed Arabic.

2.2. Chinese subword-based word segmentation

We used a Chinese subword-based word segmentation [2] that is illustrated in Figure 2. This word segmentation is composed of three steps. The first is a dictionary-based word segmentation, similiar to the default word segmentation provided by LDC. The second is a subword-based IOB tagging implemented by a CRF tagging model. The subword-based IOB tagging achieves a better segmentation than character-based IOB tagging. The third step is confidence dependent disambiguation to combine the previous two results.

The subword-based word segmentation was evaluated in both the Sighan Bakeoff and the NIST machine translation. In the second Sighan Bakeoff, the segmentation gave a higher F-score than the best published results. We also evaluated this in SMT using the NIST evaluation 2005 data, its BLEU score was 1.1% higher than that using the LDC provided default word segmentation. The results are shown in Table 2.2.

Table 2: Comparison of different Chinese word segmentations for the NIST 2005 test data

	BLEU	NIST	WER	PER	METEOR
LDC default	0.226	7.62	0.895	0.642	0.528
OURs	0.237	7.93	0.867	0.614	0.525

2.3. Japanese and Italian word segmentation

A Ngram-based (word trigrams + POS trigrams) word segmentation was used for Japanese processing. No Italian word segmentation was required.

2.4. Sentence splitting

Sentence splitting is a new technique we applied to this evaluation. We used sentence splitting to cut long sentences into short segments. We did so by automatically adding punctuation to the ASR output without punctuation, and splitting

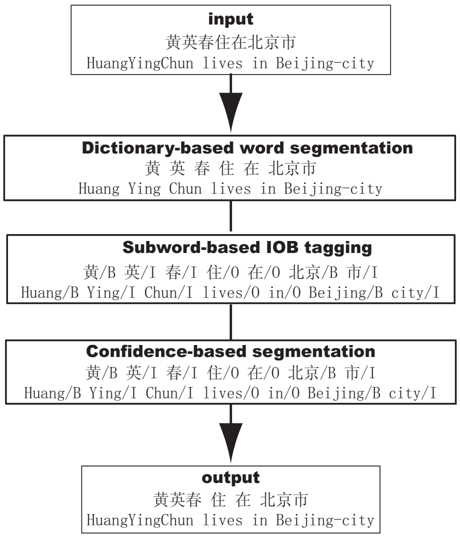
input
黄英春住在北京市
HuangYingChun lives in Beijing-city
Dictionary-based word segmentation
黄 英 春 住 在 北京市
Huang Ying Chun lives in Beijing-city
Subword-based IOB tagging
黄/B 英/I 春/I 住/O 在/O 北京/B 市/I
Huang/B Ying/I Chun/I lives/O in/O Beijing/B city/I
Confidence-based segmentation
黄/B 英/I 春/I 住/O 在/O 北京/B 市/I
Huang/B Ying/I Chun/I lives/O in/O Beijing/B city/I
output
黄英春 住 在 北京市
HuangYingChun lives in Beijing-city

Figure 2: Subword-based Chinese word segmentation

the output in the place of the added punctuation. Each segment was passed to translation engines. The final translation of the original ASR output was linearized by all the segments' translation. From the statistics, there were 1.28 segments for each sentence on average. Sentence splitting was implemented by adding punctuation using the SRI tool, "hidden-ngram."

3. Topic-dependent language model adaptation

A language model plays an important role for SMT. The effectiveness of a language model is significant if the test data happen to have the characteristics of the training data for the language models. However, this coincidence is rare in practice. To avoid this performance reduction, a topic adaptation technique is often used. We applied this adaptation technique to machine translation.

For this purpose, a "topic" is defined as clusters of bilingual sentence pairs. For bilingual sentence pair clustering, the following process is used:

1. The number of topic is the number of fixed clusters.

2. All of the sentence pairs are randomly assigned to one cluster.

3. For each cluster, language models for e and f are created using the sentence pairs that belong to each cluster.

4. For each cluster, the entropy for e and f is calculated using the language model from each cluster. Total entropy is defined as total sum of the entropies of each cluster.

Table 3: Topic adaptation for J-E translation

Model	BLEU	NIST	WER	PER
Baseline	22.17	6.68	68.09	51.72
Adapted	23.57	6.81	67.16	50.51

Table 4: Topic adaptation for C-E translation

Model	BLEU	NIST	WER	PER
Baseline	21.66	6.78	70.88	51.29
Adapted	22.77	6.96	69.75	50.75

5. Each translated sentence pair in each cluster is moved to other clusters to give the smallest total entropy.

6. Repeat above process, until the entropy reduction is smaller than a given threshold.

Topic can be defined according to the above process. In the decoding, for a source input sentence, f, a topic T is determined by maximizing $P(f|T)$. To maximize $P(f|T)$ we select cluster T that gives the largest likelihood for a given translation source sentence f.

After the topic is found, a topic-dependent language model $P(e|T)$ is used instead of $P(e)$, the topic-independent language model in the log-linear models.

The topic-dependent language models were tested using IWSLT06 data. Experimental results are shown in the tables 3 and 4. Our approach improved the BLEU score by 1.1% ∼ 1.4%.

The paper of [6] presents a detailed description for this work. As far as we know, a clustered language model was also used by [7] in this evaluation. Our work and results were similar to theirs.

In this evaluation, topic-dependent language model adaptation was used in only the TATR engine and in the translation of JE, CE and IE.

4. Subword-based translation for Chinese

It is possible that some words in the test data are not in the translation table extracted from the bilingual training corpus. There are no translations for those words. Some of these words are rare words. Some are out-of-vocabulary (OOV) words recognized by the subword-based word segmentation that can recognize Chinese numerical expressions and named entities such as place name, organization name, and person name. These new generated words cannot find corresponding translations in the translation table. For example, "长春路" is a new word generated by the segmenter if it is labeled as, "长/B 春/I 路/I". "长春路" cannot be found in the translation table, thus cannot be translated.

As a Chinese word is composed of two or more connected characters, we introduce subwords and segment a non-translated word into subwords, each of which consists of fewer characters than the original word. Even if the original word is a rare word or OOV, the resulting subwords are not, and are translatable respectively.

The subword-based translation model was trained as follows: First, we defined a subword list from the LDC corpus, consisting of the most frequent words. There were 5,000 words in the list. Second, we used an LDC-provided Chinese named entity corpus, LDC2002L27, as the bilingual corpus for training the subword translation model. We segmented Chinese sentences in the corpus into subwords, using a dictionary-based word segmentation approach. Thus, we obtained a training corpus for the translation model with subword sequences on the Chinese side and the corresponding English translation. Third, a phrase-based translation model for translating subwords was trained using the same training approach (described in the next section).

Once an OOV is found in the test data, we first apply a subword-based word segmenter to segment the OOV into a subword sequence, and then we use the subword translation model to translate the OOV. Finally we append the OOV and its translation into the translation table, so the OOV can be translated using the new translation table. By using this approach, about 95% of the OOVs can be translated.

We tested the subword-based OOV translation model using the NIST MT 2005 evaluation data, a 0.4% BLEU score increase was observed.

5. Translation engines

We used three translation engines in this evaluation: TATR, a phrase-based SMT system; HPATR3, an SMT system based on syntactic transfer; and EM by exact match. For the OPEN track, only TATR was used; for the CSTAR track, a hybrid system using three engines and *Selector* was used. See Figure 1.

5.1. TATR

TATR is a phrase-based SMT system built within the framework of feature-based exponential models:

$$Pr(e_1^I|f_1^J) = \frac{exp(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J))}{\sum_{I' e'^{I'}_1} exp(\sum_{m=1}^{M} \lambda_m h_m(e'^{I'}_1, f_1^J))}. \qquad (1)$$

The best translation, $\hat{e}_1^I$, is the maximal solution of

$$\hat{e}_1^{\hat{I}} = \max_{I, e_1^I} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \right\} \qquad (2)$$

.

where h_m are features. We used the following features.

- phrase translation probability from source to target
- inverse phrase translation probability
- lexical weighting probability from source to target
- inverse lexical weighting probability
- phrase penalty

5.2. HPATR3

HPART3 is developed from HPATR2 [8] which is a statistical MT system based on syntactic transfer. The difference between HPATR3 and HPATR2 is that HPATR3 uses a log-linear model and minimum error rate training.

The translation model of HPATR3 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: probability of source tree model, probability of target tree model, and probability of tree-mapping model. A characteristic of HPATR3 is that the syntactic transfer carried out not only word translations but also translation of multi-word sequences. Parsing hypotheses, which are multi-word sequences connected by context-free grammar rules, are created. The best hypothesis (parse tree and translation) is selected according to the models used.

Therefore, HPATR3 is an MT system that contains features of phrase-based SMT as well as syntax-based SMT.

5.3. 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. If an exact match can be achieved, the corresponding target language sentence will be used. Otherwise, the system fails to output a translation.

5.4. *Selector*

In order to select the best translation among outputs generated by multiple MT systems, we employ an SMT-based method that 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 obtained TM·LM scores of one MT output are significantly higher than those of another MT output [9]. Given an input sentence, m translation hypotheses are produced by the component MT engines ($m = 1$ for this evaluation), whereby n different TM·LM scores are assigned to each hypothesis. In order to check whether the highest scoring hypothesis is significantly better then the other MT outputs, a multiple comparison test based on the Kruskal-Wallis test is used [10]. If one of the MT outputs is significantly better, this output is selected. Otherwise, the output of the MT engine that performs best on a development set is selected.

6. Post-processing

The official submission format for the evaluation results is a case-sensitive English translation with punctuation. Because our translation engines give English translations in lowercase without punctuation, we need to recover capitalization and punctuation.

We experimented with two approaches for this purpose. The first used by using SRI LM tools [1], “disambig and hidden-ngram”. The second used in-house tools based on discriminative training. We found that our capitalization tool achieved a higher F-score result than the SRI tools, but the punctuation tool was not promising.

6.1. Punctuation using the SRI tools

As to punctuation, we used the SRI tools, “hidden-ngram,” which is based on Ngram language models. In fact, we developed an in-house punctuation tool based on maximum entropy (ME) method, where we can view a punctuation behind a word as a label to the previous word. We integrated many features into the ME-based model. However, our punctuation results were not satisfactory. The F-score was lower than that of using the SRI tools. Finally, we decided to use the SRI tools for punctuation. We are still investigating this problem.

6.2. Capitalization based on CRF++

Our capitalizer is modeled by the conditional random fields(CRF) approach. We view the problem of capitalizing lowercase words as one of labeling words with one of four tags: AL, IU, AU, or MX, that stand for all lowercase, initial uppercase, all uppercase and mixed case.

For example, the sentence, *McAdam is CEO of a British company*, is labeled as, *mcadam/MX is/AL CEO/AU of/AL a/AL British/IU company/AL.*

The CRF tagging model is defined as,

$$
\begin{aligned}
&p(T|W) = \\
&\exp\left(\sum_{i=1}^{M}\left(\sum_{k}\lambda_k f_k(t_{i-1}, t_i, W) + \sum_{k}\mu_k g_k(t_i, W)\right)\right)/Z, \qquad (3)\\
&Z = \sum_{T=t_0 t_1 \cdots t_M} p(T|W)
\end{aligned}
$$

where T is a tag sequence and W is a word sequence for tagging. f_k and g_k are unigram and bigram features, and λ and μ_k are feature’s values.

We used word features only. An example of the use of model 3 is shown in [2]. We used CRF++ [2] to train the CRF tagger.

Our capitalization model achieved higher accuracy than the SRI tools. For testing the performance of our capitalizer, we used the devset4 reference data. We removed the punctuation and lowercased the reference data, and then used our tools to recover punctuation and capitalization. We measured the results in terms of BLEU against the original reference data. We compared our tools with the SRI tools. The improvement was about 10%. The BLEU score increased from 0.81 using SRI tools to 0.827 using our in-house capitalizer.

7. Evaluation Results

As mentioned in Section 1, we participated in 14 of the 18 tests. The training data statistics for all language pairs and tracks (OPEN and CSTAR) are shown in Table 5, where “Source” stands for source language and numbers in the parentheses indicate number of distinct sentences. “word tokens” indicates a word with an attached tag. “word types” is the surface form.

The training parallel corpus size was 40,000 for CE and JE and 20,000 for AE and IE in the OPEN track. For the CSTAR track, we used 600,000 sentence pairs for training the translation model and LMs.

The language model for English was learnt on a larger set of English data than the supplied data. In addition to the 20,000 supplied data sentences, 190,000 sentences from the business domain was used. However, we found that smaller than 0.5% BLEU score increases were earned for all language translations as a result of enlarging the data.

We used GIZA++ [11] and Pharaoh [3] for training and parameter tuning. The process of training the translation models for the TATR engine was the same for all language pairs except in regard to data preprocessing.

We used the TATR translation engine for the OPEN track. We used TATR, HPATR3, EM, and *Selector* for the CSTAR track.

The decoding process was divided into several steps: (1) For a given ASR output without case and punctuation, we used the SRI tools to insert punctuation into the output. (2) The ASR output was split according to the inserted punctuation. (3) We translated each split segment separately using multiple MT engines: TATR, HPATR3 and EM. (4) We selected the best translation hypothesis for each split segment separately by using the *Selector*. This step was spared for the OPEN track. (5) We recombined all segment translations to obtain the translation output. (6) We inserted punctuation and capitalization as described in Section 6 to obtain the final English translation output.

All the results submitted in the official runs are shown in Table 7 and Table 9. The official submissions are with punctuation and capitalization. The results without punctuation and capitalization are shown in Table 8 and Table 10. In the tables, the numbers in each slot indicate the ASR output (before ”/”) and the correct transcription (after ”/”).

The results indicate the following:

- There is a 3% to 6% increase in terms of BLEU score for the correct transcription translation relative to the

[1] http://www.speech.sri.com/projects/srilm

[2] http://www.chasen.org/t̃aku/software/CRF++

[3] http://www.iccs.inf.ed.ac.uk/ pkoehn/

Table 5: Training data statistics

		#Sentences		#Word Count		#Word Tokens		#Word types	
		Source	English	Source	English	Source	English	Source	English
CE	OPEN	39,953(37,559)	39,953(39,633)	342,362	367,265	11,174	9,263	11,174	7,225
	CSTAR	678,748(399,527)	716,280(358,681)	4,606,373	5,756,026	43,273	28,851	43,271	21,809
JE	OPEN	39,953(37,173)	39,953(39,633)	398,498	367,265	13,627	9,263	11,407	7,225
	CSTAR	691,711(490,499)	651,558(444,859)	6,795,833	5,514,327	56,021	32,291	45,111	24,295
AE	OPEN	19,972(19,777)	19,972(19,880)	154,279	183,673	18,292	6,940	18,292	5,465
IE	OPEN	19,972(19,641)	19,972(19,880)	171,764	183,673	10,085	6,940	10,085	5,465

ASR output translation. Hence, the ASR error rate has a significant impact on translation.

- Using more data improves translation because the results of the CSTAR track are better than those of the OPEN track.

- Comparing spontaneous speech and read speech translation in the CE track, we found that the translation results of spontaneous speech were more erroneous than those of read speech. This is because the ASR error rate is higher for spontaneous speech recognition.

- If a higher BLEU score means the language is easier to translate, the order of languages in terms of ease of translation seems to be IE>AE>JE>CE according to the BLEU scores. Remarkably, IE and AE used fewer training data but had a higher BLEU score than JE and CE.

- Comparing Table 7 and Table 8, Table 9 and Table 10, we see that for CE and JE, the results with case and punctuation are slightly better than without case and punctuation. However, for AE and IE, the reverse is true. The results with case and punctuation are much worse than without case and punctuation. Because we recover case and punctuation after translation, this seems to prove that adding case and punctuation reduces the translation performance for AE and IE translation. Because CE and JE are difficult language pairs, the effect of adding case and capitalization is not easily observable.

- Table 9 and Table 10 show the contributions of the single MT engines for the CSTAR track. We used three translation engines and a *Selector*. We found that the *Selector* achieved a better BLEU score for JE read speech.

The ASR output is without punctuation. Before the ASR output is translated, we added punctuation to it and applied a sentence splitting technique to split the ASR output into segments. Our translation engines translated each segment and finally assembled these translations in sequence. Table 11 compares results with and without sentence-splitting. We found that the sentence splitting technique significantly improved the BLEU score for CE and JE. However, its results were slightly worse for AE and IE. Recalling the previous experiments showing that adding punctuation after translation reduced the BLEU scores for AE and IE, we feel that keeping punctuation information in training translation model is the right strategy for AE and IE. However, it is better to remove punctuation for CE and JE in training. One possible explanation is that AE and IE are similar language pairs.

Table 6: Contributions of single engines in official run submission

		TATR	HPATR3	EM
CE	spontaneous	454 (90.8%)	46 (9.2%)	0
	read	452 (90.4%)	48 (9.6%)	0
	correct	455 (91%)	42 (8.4%)	3 (0.6%)
JE	read	405 (81%)	92 (18.4%)	3 (0.6%)
	correct	408 (81.6%)	86 (17.2%)	6 (1.2%)

8. Conclusions

In this IWSLT evaluation, we used several new approaches: subword-based word segmentation, named entity recognition and translation, ME- and CRF-based punctuation and capitalization, sentence splitting, and language model adaptation. These approaches proved effective in the recent NIST machine translation evaluation; however, we didn't evaluate these approaches completely in the IWSLT task due to time limitations.

This year's system differs from last year's system in that we used a phrase-based statistical machine translation system, TATR. This system is still in the preliminary stages of development. Many important models such as the distortion model are not implemented yet. A simple position-dependent parameter was used in the decoding to represent the distortion. We expect to improve this system in our future work.

We used the 1-best translation for the ASR track in this evaluation. We could achieve a better score if we used N-best or lattice translation.

Table 7: Translation results in the OPEN track (with case and punctuation)

	BLEU4	NIST	METEOR	WER	PER
CE spontaneous speech	0.1591/0.206	4.9696/5.8613	0.4117/0.487	0.7291/0.6837	0.5851/0.5314
CE read speech	0.1775/0.206	5.2286/5.8613	0.4336/0.487	0.7147/0.6837	0.5705/0.5314
JE read speech	0.1899/0.2122	5.5915/5.9494	0.4574/0.49	0.6984/0.6657	0.5458/0.5182
AE read speech	0.2117/0.2365	5.9216/6.3521	0.4867/0.5224	0.6354/0.6112	0.5272/0.4986
IE read speech	0.2989/0.3763	6.8985/8.1318	0.5744/0.663	0.55/0.4738	0.4641/0.3901

NOTE: the numbers indicate the translatios of the ASR output (before "/") and the correct transcription (after "/")

Table 8: Translation results in the OPEN track (without case and punctuation)

	BLEU4	NIST	METEOR	WER	PER
CE spontaneous speech	0.1615/0.2123	5.3592/6.3848	0.4114/0.4862	0.7481/0.6946	0.5746/0.5105
CE read speech	0.1772/0.2123	5.6649/6.3848	0.4323/0.4862	0.7290/0.6946	0.5583/0.5105
JE read speech	0.1832/0.2077	5.9428/6.3325	0.4569/0.4893	0.7219/0.6826	0.5370/0.5018
AE read speech	0.2164/0.2463	6.3959/6.8893	0.4869/0.5229	0.6406/0.6105	0.5055/0.4734
IE read speech	0.3194/0.412	7.4724/8.9027	0.5739/0.6625	0.5342/0.4450	0.4265/0.3415

Table 9: Translation results in the CSTAR track (with case and punctuation)

	BLEU4	NIST	METEOR	WER	PER
CE spontaneous speech					
Selector	0.2008/0.2654	5.4009/6.5274	0.4502/0.5425	0.6994/0.6380	0.5629/0.5003
TATR	0.2002/0.2635	5.4077/6.5485	0.4498/0.5427	0.7033/0.6425	0.5660/0.5034
CE read speech					
Selector	0.2155/0.2654	5.6857/6.5274	0.4787/0.5425	0.6733/0.6380	0.5443/0.5003
TATR	0.2189/0.2635	5.7302/6.5485	0.4792/0.5427	0.6748/0.6425	0.5463/0.5034
JE read speech					
Selector	0.2487/0.2861	6.2778/6.8327	0.5039/0.5536	0.6569/0.6104	0.5118/0.47
TATR	0.2463/0.2875	6.2447/6.8588	0.5018/0.5518	0.6617/0.6180	0.5146/0.4716
HPATR3	0.2177/0.2597	5.852/6.6586	0.4833/0.5308	0.6998/0.6317	0.5603/0.5057

Table 10: Translation results in the CSTAR track (without case and punctuation)

	BLEU4	NIST	METEOR	WER	PER
CE spontaneous speech					
Selector	0.2039/0.2751	5.8205/7.086	0.4492/0.5419	0.7129/0.6384	0.5482/0.4765
TATR	0.2053/0.2745	5.852/7.1355	0.4488/0.5420	0.7170/0.6430	0.5500/0.4778
CE read speech					
Selector	0.2214/0.2751	6.1453/7.086	0.4783/0.5419	0.6813/0.6384	0.5304/0.4765
TATR	0.2254/0.2745	6.1993/7.1355	0.4787/0.5420	0.6833/0.6430	0.5306/0.4778
JE read speech					
Selector	0.2466/0.2867	6.7157/7.3021	0.5032/0.5529	0.6726/0.6191	0.4994/0.4533
TATR	0.2438/0.2851	6.6367/7.3166	0.5011/0.5510	0.6782/0.6295	0.50340.4564/
HPATR3	0.2131/0.2555	6.2587/7.1613	0.4825/0.5300	0.7237/0.6472	0.5609/0.5003

Table 11: Comparison of results with and without sentence splitting

	BLEU4	NIST	METEOR	WER	PER
CE spontaneous speech					
with	0.1591	4.9696	0.4117	0.7291	0.5851
without	0.1551	4.9322	0.4095	0.7382	0.5871
CE read speech					
with	0.1775	5.2286	0.4336	0.7147	0.5705
without	0.1756	5.2115	0.4336	0.7229	0.5719
CE correct					
with	0.206	5.8613	0.487	0.6837	0.5314
without	0.2051	5.8468	0.4876	0.6944	0.5316
JE read speech					
with	0.1899	5.5915	0.4574	0.6984	0.5458
without	0.1817	5.4639	0.4532	0.7228	0.5524
JE correct					
with	0.2122	5.9494	0.49	0.6657	0.5182
without	0.2023	5.8312	0.4826	0.6925	0.5225
AE read speech					
with	0.2117	5.9216	0.4867	0.6354	0.5272
without	0.2122	5.9287	0.4874	0.6345	0.5263
AE correct					
with	0.2365	6.3521	0.5224	0.6112	0.4986
without	0.2384	6.3691	0.5221	0.6100	0.4976
IE read speech					
with	0.2989	6.8985	0.5744	0.55	0.4641
without	0.2991	6.9066	0.5713	0.5551	0.4627
IE correct					
with	0.3763	8.1318	0.663	0.4738	0.3901
without	0.3774	8.1429	0.6601	0.4766	0.3880

9. References

[1] M. Paul, T. Doi, Y. Hwang, K. Imamura, H. Okuma, and E. Sumita, "Nobody is perfect: ATR's hybrid approach to spoken language translation," in *Proc. of the International Workshop on Spoken Language Translation*, Pittsburgh, USA, 2005.

[2] R. Zhang, G. Kikui, and E. Sumita, "Subword-based tagging by conditional random fields for chinese word segmentation," in *Companion volume to the proceedings of the North American chapter of the Association for Computational Linguistics (NAACL)*, 2006, pp. 193–196.

[3] T. Buckwalter, "Buckwalter arabic morphological analyzer version 1.0." Linguistic Data Consortium" Technical report LDC2002L49, 2002. [Online]. Available: http://www.ldc.upenn.edu/

[4] F. Sadat and N. Habash, "Morphological preprocessing scheme combination for statistical MT," in *Proceedings of COLING-ACL*, 2006.

[5] ——, "Arabic preprocessing schemes for statistical machine translation," in *Proceedings of the North American chapter of the Association for Computational Linguistics (NAACL)*, 2006.

[6] H. Yamamoto and E. Sumita, "Online language model task adaptation for statistical machine translation (in Japanese)," in *FIT2006*, Fukuoka, Japan, 2006, pp. 131–134.

[7] A. Mauser, R. Zens, E. Matusov, S. Hasan, and H. Ney, "The rwth statistical machine translation system for the iwslt 2006 evaluation," in *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2006.

[8] 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.

[9] 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.

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

[11] F. J. Och and H. Ney, "A systematic comparison of various statistical alignment models," *Computational Linguistics*, vol. 29, no. 1, pp. 19–51, 2003.