The NICT/ATR Speech Translation System for IWSLT 2007

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

ATR Spoken Language Communication Research Labs
Hikaridai 2-2-2, Keihanna Science City, 619-0288 Kyoto
Andrew.Finch@atr.jp

Abstract

This paper describes the NiCT-ATR statistical machine translation (SMT) system used for the IWSLT 2007 evaluation campaign. We participated in three of the four language pair translation tasks (CE, JE, and IE). We used a phrase-based SMT system using log-linear feature models for all tracks. This year we decoded from the ASR n-best lists in the JE track and found a gain in performance. We also applied some new techniques to facilitate the use of out-of-domain external resources by model combination and also by utilizing a huge corpus of n-grams provided by Google Inc.. Using these resources gave mixed results that depended on the technique also the language pair however, in some cases we achieved consistently positive results. The results from model-interpolation in particular were very promising.

1. Introduction

Phrase-based statistical machine translation approaches continue to dominate the field of machine translation. All of the systems for each of the languages pairs we submitted results for differ in important respects from other systems, however they are all based around a fairly typical phrased-based machine translation system built within the framework of a feature-based exponential model containing 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
- Language model probability
- Simple distance-based distortion model
- Word penalty

The basic framework within which all the systems were constructed is shown in Figure 1, and the corresponding overview of the translation process is shown in Figure 1. With the exception of the experiments where factored models were required, the decoder used for the training and decoding of the test data was the in-house multi-stack phrase-based decoder CleopATRa.

This paper is constructed as follows, firstly for each language pair we provide a description of the components of the system that are specific for that language pair. This description includes details of any segmentation, external resources and any specific modelling techniques that were employed. Next, we detail those parts of the process that are common to all systems. In particular we focus on the techniques we used to utilize a large corpus of word n-grams provided by Google Inc. The next sections present and discuss our experimental results, and finally we conclude and propose avenues for future research.

2. Japanese-English

2.1. Corpora

In addition to the supplied corpus, we also drew on resources from the following corpora:

- The Tanaka corpus (203K sentence pairs)
- The Yomiuri News corpus (202K sentence pairs)
- The SLDB corpus (72K sentence pairs)
- The Chinese Olympic corpus included in the Chinese-LDC (Code: 2004-863-009) (104K sentence pairs)

2.2. Pre-processing

The data was segmented using the publicly available Chasen tool.

2.3. Training data selection

Before training the MT system, we reduce the size of the additional corpora by extracting only sentences which are 'relevant' to the task. We perform the selection using language model perplexity with reference to the supplied corpus as follows:

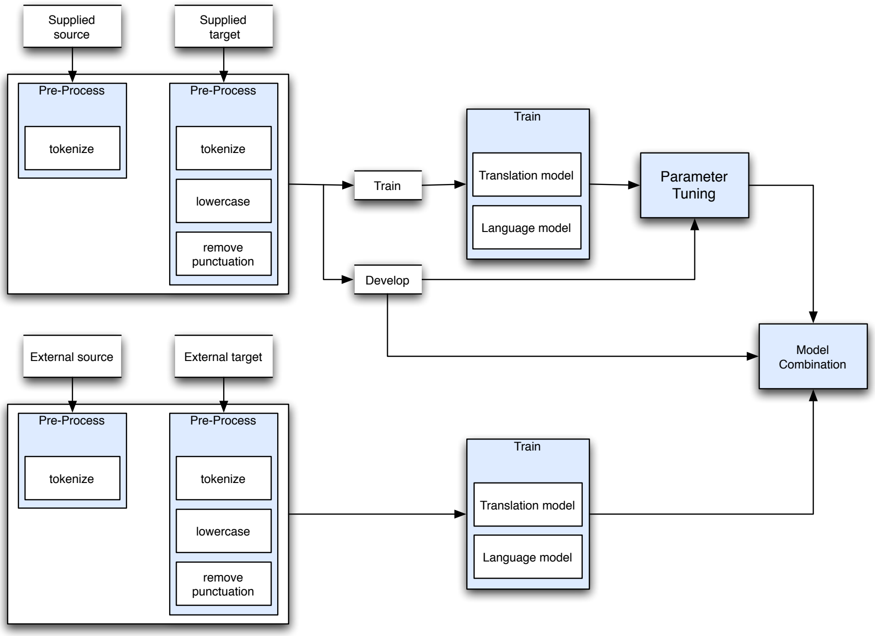

Figure 1: The training process of our translation systems

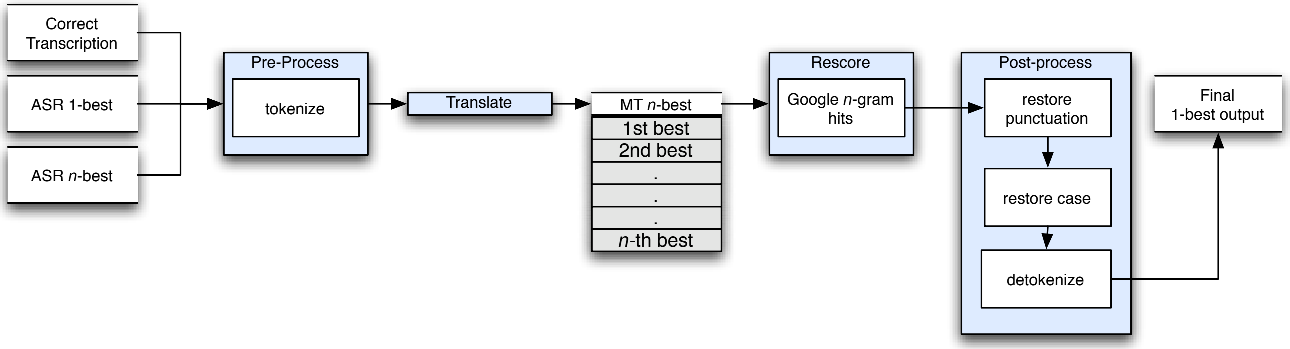

Figure 2: The translation process of our translation systems

1. An English language model (word tri-grams) was created using the supplied corpus.

2. The sentence perplexity with respect to the language model of each sentence in the additional corpus (Tanaka corpus, Yomiuri News corpus, SLDB corpus, and Beijing Olympic corpus included in Chinese-LDC) was calculated

3. Only those sentences for which the perplexity was lower than 100 were used as training sentences.

After the selection process we were left with 40K sentences from the supplied corpus and 117K additional sentences from the external corpora, giving us a total of 157K sentences for training.

2.4. Modeling Issues

Word-trigram language models were used, these were smoothed using Knesser-Ney discounting. In addition, topic-dependent models were constructed [1]. We built bilingual cluster-based models from 157K bilingual training sentence pairs. The sentence pairs were clustered into 10 sub-corpora. These sub-corpora intutitively represent sub-domains of the main corpus. The motivation behind this strategy was to build models specific to these sub-domains and then predict the sub-domain of the text to be translated, and use the appropriate model for the translation process. A strong improvement was demonstrated using this technique for all language pairs in the IWSLT06 evaluation campaign. In this year's campaign we only apply this technique to the Japanese-English task.

2.5. ASR n-Best Decoding

For the Japanese-English data track, the decoding was performed directly from the ASR n-best lists (for all experiments a value of n=20 was used) rather than from the 1-best ASR hypothesis. To do this, the ASR scores were added to the machine translation scores in a log-linear fashion with weights. The translation scores of n-best hypotheses from the machine translation were then combined with the weighted ASR scores and the n-best translation hypotheses re-ranked. The weights for the ASR scores were trained independently from the weights of the translation model, on development data and were optimized with respect to the same BLEU score used to optimize the MT decoder's parameters during minimum error rate training. Decoding directly from the confusion network was also tried out on the development data. This gave approximately the same level of improvement as decoding from the n-best list and the latter approach was selected because of its simplicity an also because of it's flexibility. For example it, permitted the data to be segmented before being decoded. If the confusion network is decoded directly we must accept the segmentation provided by the ASR system, or devise a method for re-segmenting the tokens in the confusion network.

3. Chinese-English

3.1. Corpora

We used the supplied corpus in combination with the Beijing Olympic Corpus, and other corpora provided by the LDC. These corpora and their respective sizes are shown in Table 1.

3.2. Lemmatization

Data sparseness is one of the key factors that degrade statistical machine translation (SMT). Especially for a translation task like IWSLT, where collecting a large amount of in-domain data is very expensive. One method to reduce the translation degradation caused by this approach is by using lemmatization [2]. Lemmatization is shallow morphological analysis, which uses single a lexical entry to replace a whole range of derived inflected words. For example, the three words: "doing", "did" and "done", can be replaced by one word: "do". In fact, they should all be mapped to the same Chinese target word during alignment. It is easy to see that as a result, the process reduces the number of types observed in the data, thereby easing the problems associate with sparse data, and in Chinese at least we expect the process to preserve as much of the semantic information as possible.

We used Moses to implement the method. Moses is a publicly-available state-of-the-art decoder for SMT. It is an extension of Pharaoh (Koehn et al., 2003), and supports factored training and decoding. Our idea can be easily implemented with the functionality offered by Moses. We feed Moses English words with two factors: the surface word from and the lemma. The only difference in training with lemmatization from that without is the alignment factor. The former uses Chinese surface words and English lemmas as the alignment factor, but the latter uses Chinese surface words and English surface words. Therefore, the lemmatized English is only used in the word alignment stage of the training. All the other aspects of the training process are the same for both the lemmatized translation training and non-lemmatized training.

3.3. Translation model combination

Linear interpolation of translation models has been shown to be effective in machine translation [2, 3]. In this campaign we apply this approach as the main means of integrating models built from the external resources with the primary models built from the supplied corpus. More formally, we use the following equation for model combination:

$$p(e|f) = \alpha_1 p_1(e|f) + \alpha_2 p_2(e|f)$$

where p_1 and p_2 are two models to be integrated, and the weight α_1 and α_2 must sum to unity.

We did not use automatic optimization methods to select the α_1 and α_2. Instead, we hand-selected the values by evaluating the performance of multiple runs on the development data. We consider this approach reasonable since the system's performance was fairly insensitive to changes in these parameters.

3.3.1. Translation models used

All of the bilingual data that was used for training the translation model is shown in Table 1. The first corpus listed, "IWSLT07 supplied corpus", is the organizer-provided training data for IWSLT 2007. Since the Chinese Olympic data has been drawn from travel domain, we treated it as if from the same source as the IWSLT 2007 data. This data was treated differently from the other LDC data, the last resource in the table, which we considered to be out-of-domain with respect to the IWSLT07 supplied corpus.

The final translation models were obtained by the following steps:

- Merge the IWSLT07 supplied corpus and the Olympic corpus
- Train a translation model, $m1$, using the above data
- Lemmatize the above data
- Train a translation model, $m2$, using the lemmatized data
- Linearly interpolate models $m1$ and $m2$ to yield a model, $m3$.
- Train a translation model, $m4$, using the LDC data.
- Linearly interpolate models $m3$ and $m4$ to yield the final model, $m5$.

Table 1: Training data for the CE translation model

source	# of sentence pairs	Description
IWSLT07 supplied corpus	40K	provided by IWSLT 2007
Chinese Olympic corpus	50K	part of the CLDC Corpus 2004-863-009
LDC	2.5 M	LDC corpus (LDC2002T01, LDC2003T17 LDC2004T07 LDC2004T08 LDC2005T06 and LDC2005T10)

Table 2: Experiments

TM	BLEU
provided data	46.65
provided+LDC	49.70
provided+lemmatized+LDC	50.48
provided+Olympic+lemmatized+LDC	51.78
provided+Olympic+lemmatized+LDC+MERT	57.32

We used equal weights for interpolating the all of the models, with the exception of the model $m3$ built from the LDC corpus which was weighted with a weight of 0.3 (and therefore 0.7 for $m4$).

3.4. Experiments

The results shown in Table 3.4 were from experiments made on the development data; the test data of IWSLT 2006. The results proved the effectiveness of both of our methods: model interpolation and lemmatization. In the table, the first column describes the training data and the details of interpolated models and lemmatization used. The second column gives the corresponding BLEU scores. We found that in every case where we used lemmatization and model interpolation, the BLEU scores were higher than without. We used minimum-error rate training MERT in the last experiment only, shown by the last line. The development data and test data that was used for this were from those used in the IWSLT2004 campaign.

4. Italian-English

4.1. Corpora

We made use of all supplied IE CSTAR data (20k sentence pairs), and of EUROPARL data (940k sentence pairs). Preliminary experiments on the EUROPARL data showed best results by retaining only sentences pairs with a length ratio greater than 0.85 (around 940,000 pairs in total). These pilot experiments showed that interpolating phrase tables created using the supplied corpus with those created from EUROPARL data gave respectable improvements on the dev5a development set. Unfortunately these improvements did not transfer to dev5b. Therefore only data from the supplied corpus was used for phrase table estimation of the primary system. However, the EUROPARL data proved useful for language modelling purposes. We interpolated LM's built on the supplied corpus with language models built on the EUROPARL data, used the resulting model for translation. As data in the source language of the supplied corpus is lowercased and without punctuation, the source data of the EUROPARL corpus was transformed to match this. The target data was also lowercased, and punctuation was removed.

4.2. Methodology

Alignments were obtained by using GIZA++, and minimum error rate training with respect to the BLEU metric was performed by using the provided development set dev5b (995 sentences with 1 reference translation for each source sentence). Language models (built with modified Kneser-Ney discounting and lower-order interpolation) were made using the SRILM toolkit. The decoding was performed using a tri-gram language model with no limit on the maximum distortion distance.

5. Punctuation, Case and Tokenization

The format of the official submission of the data for evaluation is case-sensitive and with punctuation. Based on the results of a series of preliminary experiments on Italian-English data that showed slightly higher performance from re-punctuated/re-capitalized test data, we elected to recover the capitalizion and punctuation in a separate post-processing step. This efficacy of this approach could depend on the particular language involved, our experiments only addressed one language pair, and it is our intention to investigate this in further research. The experiments also showed that reasonably large differences in BLEU score can arise from differing punctuation/capitalization schemes indicating that this part of the task is an important component. Differing tokenization schemes were adopted depending on the language involved these are described in the respective section for the language pair.

5.1. Pre-processing

All of the training and development data was converted to lowercase and punctuation stripped before training began. Punctuation characters that were part of works (for example apostrophes) were left in the data. The English data was also tokenized using an in-house tokenizer that tokenized the text in a very similar form to the UPENN tree-bank tokenization scheme.

5.2. Post-processing

Since the models were trained on unpunctuated lowercase data, the system's output required this information to be restored for evaluation.

We experimented with two approaches for this purpose. In the first method we used two of the tools provided in the SRI Language Modelling Toolkit [1], "disambig" and hidden-ngram". The second method employed in-house tools based on discriminative training methods. We found that the in-house tool capitalization tool achieved a higher F-score result than the SRI tools, but for punctuation the in-house tool's performance was less promising.

As a consequence, for the punctuation step, we used the SRI tools', "hidden-ngram" application which is based on n-gram language modelling techniques. The in-house punctuation tool punctuation tool based on a maximum entropy (ME) tagging method, where we view a punctuation behind a word as the label of the previous word. We incorporated a considerable number of features into the ME-based model, however in our experiments this model was outperformed in terms of F-score by the SRI tool.

The main differences from the approach taken in last year's evaluation campaign [4] were:

- Usage of out-of-domain resources (the English part of the EUROPARL corpus)
- Usage of a linear combination of in- and out-of-domain models (see Figure 5.2). We used the following interpolation weights for all experiments: 0.1*$LM_{europarl}$ + 0.9*$LM_{supplied\_corpus}$
- The post-processing was applied after rescoring, based on the results of experiments conducted on the development sets (see Table 4).

5.3. CRF-based Capitalization

Our capitalizer is modeled by conditional random fields (CRF). We view the problem of capitalizing lowercase words as labeling the words with one of four tags: AL, IU, AU, 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 expressed by the following equation:

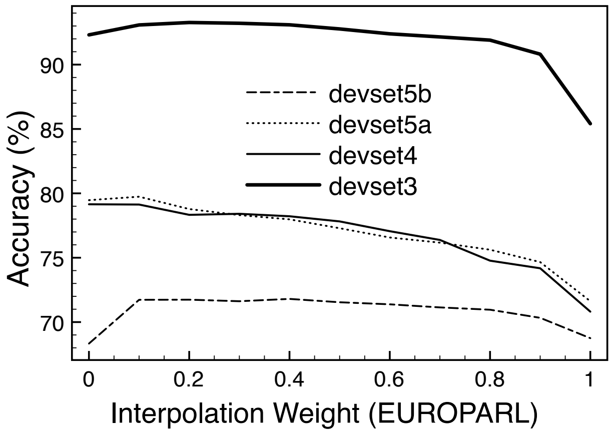

Figure 3: The effect of integrating models built from external resources for case and punctuation restoration

$$p(T|W) = \exp\left(\sum_{i=1}^{M}\left(\sum_{k}\lambda_k f_k(t_{i-1},t_i,W) + \sum_{k}\mu_k g_k(t_i,W)\right)\right)/Z,$$
$$Z = \sum_{T=t_0t_1\cdots t_M} p(T|W) \qquad (1)$$

Where T is a tag sequence and W is a word sequence for tagging. f_k and g_k are unigram and bigram features. λ and μ_k are feature values.

We used lexical features only. An example of the use of model 1 is given in [5]. We used the publicly available CRF++ [2] toolkit to train the CRF tagger.

Our model achieved higher accuracy than the SRI tools: for the IWSLT task, the improvement in accuracy was about 10% using this year's development (devset4 reference) data. The BLEU score increased from 0.81 using SRI tools to 0.827 using our in-house capitalizer. Therefore we used the in-house capitalization tool for all of the experiments in this year's evaluation.

6. Hit-rate-based Skip n-gram Rescoring

This section describes the re-scoring of the statistical MT decoder hypotheses based on skip n-gram counts extracted from a large-scale corpus consisting of collections of web-pages.

In order to handle very large amounts of training data to build language models, recent research focuses on *distributed language modeling* that use a two-pass approach to store corpora in suffix arrays and serve raw counts [6, 7] or a single-pass approach that provides smoothed probabilities

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

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

using simple smoothing techniques [8]. Although such approaches are to be preferred when available, the computational and hardware requirements are still immense and not always practicable.

In order to make use of very large training corpora with fewer resources, we use a method based on n-gram occurrence counts . The *hit-rate* of a word sequence is defined to be:

$$HitRate(w_1^L) = \sum_{i,j;i<j} \delta(w_i^j) \tag{2}$$

$$\delta(w_i^j) = \begin{cases} 1 & : \quad f(w_i^j) > 0 \\ 0 & : \quad f(w_i^j) = 0 \end{cases}$$

The *hit-rate* counts can be easily calculated, even for very large training corpora like the *Web-Corpus* introduced in Section 6.1. For the IWSLT experiments, we calculated the *hit-rate* feature for *skip- n-grams* (cf. Section 6.2) and applied it to the re-scoring of n-best translation hypotheses produced by the statistical decoder as described in Section 6.3.

6.1. Corpus

For the experiments described in Section 6.4, we used "*Web 1T 5-gram Version 1*" corpus provided by LDC [9]. This data set, contributed by Google Inc., consists of 1 trillion word tokens of text from publicly accessible Web pages. It contains English word n-grams and their observed frequency counts where the length of the n-grams ranges from unigrams (single words) to five-grams. For the experiments described in Section 6.4, we used only the 5-gram and 4-gram counts.

6.2. Skip n-grams

Skip n-grams are sequences of n words with one or more holes at any location except for the first word.

$$\omega_1^L = (w_1, \dots, w_L); w_i =' *', i \in \{2, \dots, L\} \tag{3}$$

For example, given a 5-gram ω = (w_1,w_2,w_3,w_4,w_5), the following skip n-grams can be extracted:

skip-4grams	(w_1,*,w_3,w_4,w_5) (w_1,w_2,*,w_4,w_5) (w_1,w_2,w_3,*,w_5) (w_1,w_2,w_3,w_4,*)
skip-3grams	(w_1,*,*,w_4,w_5) (w_1,*,w_3,*,w_5) (w_1,*,w_3,w_4,*) (w_1,w_2,*,*,w_5) (w_1,w_2,*,w_4,*) (w_1,w_2,w_3,*,*)
skip-2grams	(w_1,w_2,*,*,*) (w_1,*,w_3,*,*) (w_1,*,*,w_4,*) (w_1,*,*,*,w_5)

In order to obtain the hit-rate of a skip n-gram in a sequence of words, these holes are treated as wildcards (that match any single word), and the skip n-gram is matched even if the respective parts of the word sequence differ.

6.3. Hit-rate-based Re-scoring

The algorithm to re-score translation hypotheses based on the *hit-rate* of skip n-grams is given in Figure 4. For each input sentence, the n-best translation hypotheses are generated by a statistical decoder and a score S_D based on various statistical models is assigned. The re-scoring algorithm calculates the hit-rate for all skip n-grams contained in each hypothesis and linearly combines the decoder score with the respective hit-rates obtaining a new score S_R.

$$\begin{aligned} S_R(hyp) \quad &= \quad \alpha_D * S_D \\ &\quad + \sum_{i=k,\dots,l} \alpha_i * \text{HitRate}_i(hyp) \end{aligned} \tag{4}$$

The respective weights α_i can be optimized on a given development set. For each n-best list, the translation hypothesis with the highest S_R is selected as the translation output.

```
proc RESCORE( $NbestFile, NgramFile, \alpha_D, \alpha_{k,\dots,l}$ ) ;
begin
    (* read translation hypotheses from file *)
    $NbestList \leftarrow$ read-file($NbestFile$) ;
    for each $hyp$ in $NbestList$ do
        $S_D(hyp) \leftarrow$ getDecoderScore($hyp$) ;
        for each $HypSkipNgram$ in getSkipNgram($hyp$) do
            $HitRate_{HypSkipNgram} \leftarrow 0$ ;
        od ;
    od ;
    (* read NGRAM counts from file *)
    for each $ngram$ in read-file($NgramFile$) do
        for each $SkipNgram$ in getSkipNgram($ngram$) do
            $HitRate_{SkipNgram} \leftarrow HitRate_{SkipNgram} + 1$ ;
        od ;
    od ;
    (* rescore hypotheses *)
    for each $hyp$ in $NbestList$ do
        $S_R(hyp) \leftarrow \alpha_D * S_D(hyp)$
                        $+ \sum_{i=k,\dots,l} \alpha_i *$ getHitRate($hyp, i$) ;
    od ;
    $BestHyp \leftarrow$ max arg$_{hyp}$($S_R(hyp)$) ;
    return( $BestHyp$ ) ;
end ;
```

Figure 4: RESCORE algorithm

6.4. Experiments

The hit-rate-based re-scoring using skip n-grams was applied to the Italian-English translation task. The decoder translated the 1-best recognition result and the 1000-best translation hypotheses were produced for each sentence. These hypotheses were re-scored using the method described above and the hypothesis with the highest score after re-scoring was selected as the final translation. The translation quality was evaluated for the development sets *IE_dev5a* and *IE_dev5b* using the

BLEU, NIST, and METEOR metrics [3].

In the first step, we investigated the dependencies of the proposed method concerning the size N of the N-best list and the weight W for the linear interpolation of 5-gram and skip 4-gram hit-rate counts. Figure 6.4 illustrates the effects of varying N and W for both, IE_dev5a and IE_dev5b, development sets where the evaluation is carried out without case and punctuation information. The results show that for the BLEU metric an N-best list of size 40 matching only 5-grams (W=1.0) performed best for both development sets. For NIST, the largest improvement was achieved for IE_dev5a with N=30 and W=1.0, but almost no improvement was achieved for IE_dev5b. For METEOR, larger improvements could be achieved when taking into account the skip-4grams gaining 1.7 points for N=20 and W=0.3 for IE_dev5a and 0.9 points for N=1000 and W=0.7 for IE_dev5b.

Table 3 compares the results of the proposed method to the baseline method that selects the translation hypothesis with the highest decoder score. The re-scoring method outperforms the baseline method for all evaluation metrics gaining 1.5 / 0.4 points in BLEU, 13.6 / 0.2 points in NIST, and 1.6 / 0.9 points in METEOR for the *IE_dev5a* / *IE_dev5b* data sets, respectively.

Table 3: Rescoring Effects

data	rescoring	BLEU	NIST	METEOR
IE_dev5a	no	0.4288	9.1800	0.6944
	yes	**0.4434**	**9.3165**	**0.7110**
IE_dev5b	no	0.2056	5.4001	0.5265
	yes	**0.2089**	**5.4023**	**0.5351**

The n-gram-count corpus used for re-scoring also contains case and punctuation information. Using the IWSLT development sets *IE_dev5a* and *IE_ dev5b*, we investigated empirically which of the following combinations of the re-scoring and the post-process steps is most effective.

RPC *re-scoring before punctuation/case post-processing*: all skip n-grams were lower-cased and punctuation marks were treated as wildcards. for calculating the n-gram-hit-rate for each case/punc-insensitive translation hypothesis.

PRC *re-scoring after punctuation, but before case insertion*: In the first step, punctuation was inserted and case-insensitive skip n-grams were matched against the translation hypotheses. Case information was added after the re-scoring step.

PCR *re-scoring after punctuation/case post-processing*: Punctuation and case information were added to the translation hypotheses before the re-scoring step and hit-rate was calculated using case/punctuation sensitive skip n-grams.

The results summarized in Table 4 show that the re-scoring method also outperforms the baseline method for all evaluation metrics when the evaluation is carried out case/punctuation-sensitive. In total, our method produced a 1.0 / 0.9 points gain in BLEU, 8.6 / 13 points in NIST, and 1.9 / 0.6 points in METEOR for the *IE_dev5a* / *IE_dev5b* data sets, respectively. Based on these results, we selected the RPC method for the final run submissions, because the test set of IWSLT 2007 was drawn from the same corpus as the *IE_dev5b* data set.

Table 4: Re-scoring vs. Post-processing

data	rescoring	BLEU	NIST	METEOR
IE_dev5a	(none)	0.3643	8.1823	0.6887
	RPC	0.3739	8.2392	**0.7056**
	PRC	0.3663	8.1029	0.6911
	PCR	**0.3746**	**8.2680**	0.6994
IE_dev5b	(none)	0.1569	4.6345	0.5121
	RPC	**0.1660**	**4.7671**	**0.5181**
	PRC	0.1621	4.5340	0.4936
	PCR	0.1641	4.6310	0.5172

6.5. Translation Task Dependency

In addition to the Italian-English translation task, we also verified the effectiveness of the re-scoring method for the Japanese-English and Chinese-English translation tasks and the case/punctuation-sensitive evaluation results are summarized in Table 5.

Table 5: Rescoring Effects on JE and CE

data	rescoring	BLEU	NIST	METEOR
JE_dev3	no	**0.5793**	9.5847	**0.7437**
	yes	0.5643	**9.7990**	0.7437
CE_dev5	no	0.2310	5.9020	0.4945
	yes	**0.2388**	**6.2854**	**0.5146**

For Japanese-English, the baseline system results for most of the automatic evaluation metrics couldn't be improved. Moreover, due to the unavailability of the Chinese challenge task test data, the Chinese test set was changed by the organizers at short notice. Unfortunately, there was not enough time to validate the effects of the re-scoring method on the *CE_dev3* devset which was taken from the same domain as the new test set. For the above reasons, we decided to submit the runs *without* re-scoring as the primary runs for the Japanese-English and Chinese-English translation tasks.

7. Conclusions

The work for this year's evaluation campaign has focussed on the task of effectively utilizing external out-of-domain resources to support the supplied in-domain corpus. Overall

[3] For the automatic evaluation, 7 reference translations were available for *IE_dev5a* and 1 reference translation was available for *IE_dev5b*.

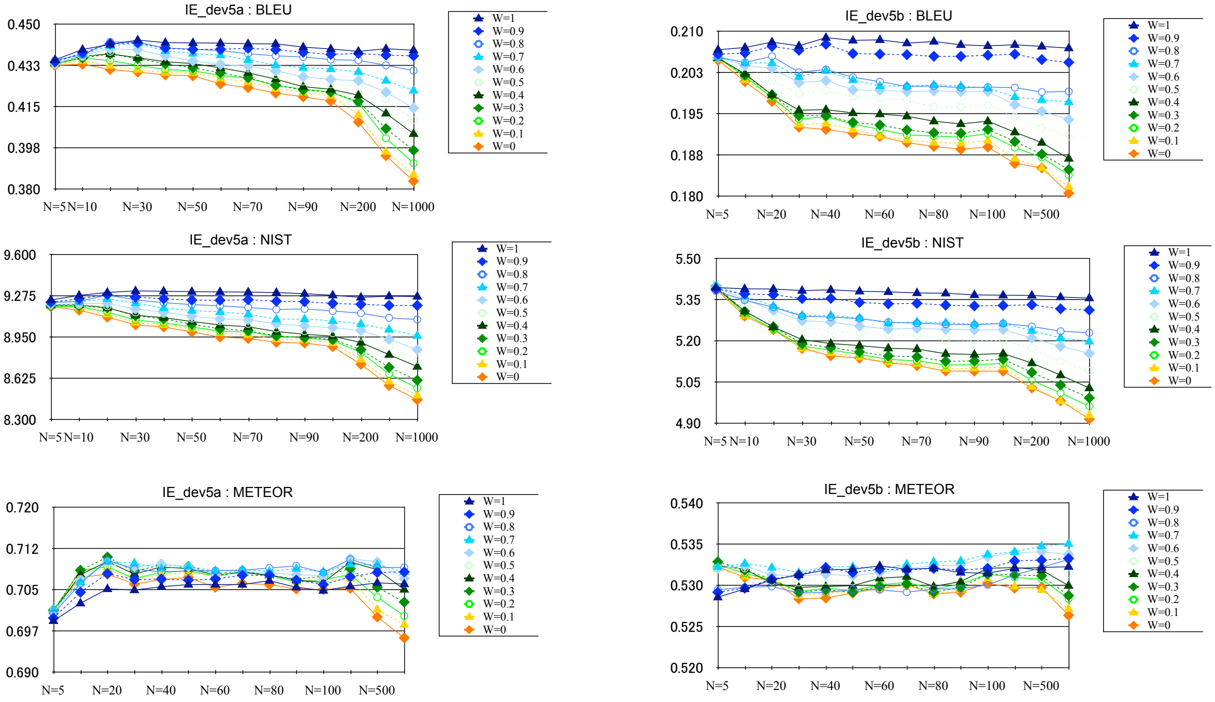

Figure 5: n-best and Weight Dependency

our experiments show that these corpora combined with the techniques we have applied are very useful, although in some cases the addition of out-of-domain data degraded system performance. It is therefore clear that we need to develop and refine the techniques further in order to exploit the external resources to the full. We also observed that the pre- and post-procesing tasks related to handing case, punctuation and segmentation can have a large impact on the automatic evaluation scoring, and it is important to improve these components alongside the machine translation component for future evaluations. Furthermore, improvements along these lines would have had a knock-on effect on the re-scoring process since more n-gram hits could be obtained thereby increasing the reliability of the selection process.

8. References

[1] H. Yamamoto and E. Sumita, “Bilingual cluster based models for statistical machine translation,” in *Proceedings of EMNLP*, 2007.

[2] R. Zhang and E. Sumita, “Boosting statistical machine translation by lemmatization and linear interpolation,” in *Proc. of the ACL: companion volume*, 2007, pp. 181–184.

[3] G. Foster and R. Kuhn, “Mixture-model adaptation for SMT,” in *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic, June 2007.

[4] R. Zhang, H. Yamamoto, M. Paul, H. Okuma, K. Yasuda, Y. Lepage, E. Denoual, D. Mochihashi, A. Finch, and E. Sumita, “The NiCT-ATR Statistical Machine Translation System for IWSLT 2006,” in *Proc. of the International Workshop on Spoken Language Translation*, Kyoto, Japan, 2006, pp. 83–90.

[5] R. Zhang, G. Kikui, and E. Sumita, “Subword-based tagging by conditional random fields for chinese word segmentation.” in *HLT-NAACL*, 2006.

[6] Y. Zhang, A. S. Hildebrand, and S. Vogel, “Distributed language modeling for n-best list re-ranking,” in *Proc. of the EMNLP*, Sydney, Australia, 2006, pp. 216–223.

[7] A. Emami, K. Papineni, and J. Sorenson, “Large-sclale distributed language modeling,” in *Proc. of the ICASSP*, Honolulu, USA, 2007, pp. IV–37 – IV–40.

[8] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, “Large language models in machine translation,” in *Proc. of the EMNLP*, Prague, Czech, 2007, pp. 858–687.

[9] *Web 1T 5-gram Version 1*, Linguistic Data Consortium, 2006, http://www.ldc.upenn.edu/Catalog/Catalog Entry.jsp?catalogId=LDC2006T13.