The NICT Translation System for IWSLT 2010

Chooi-Ling Goh, Taro Watanabe, Michael Paul, Andrew Finch, Eiichiro Sumita

Language Translation Group
MASTAR Project
National Institute of Information and Communications Technology
Kyoto, Japan
{chooiling.goh,taro.watanabe,michael.paul,andrew.finch,eiichiro.sumita}@nict.go.jp

Abstract

This paper describes NICT's participation in the IWSLT 2010 evaluation campaign for the DIALOG translation (Chinese-English) and the BTEC (French-English) translation shared-tasks.

For the DIALOG translation, the main challenge to this task is applying context information during translation. Context information can be used to decide on word choice and also to replace missing information during translation. We applied discriminative reranking using contextual information as additional features. In order to provide more choices for re-ranking, we generated n-best lists from multiple phrase-based statistical machine translation systems that varied in the type of Chinese word segmentation schemes used. We also built a model that merged the phrase tables generated by the different segmentation schemes. Furthermore, we used a lattice-based system combination model to combine the output from different systems. A combination of all of these systems was used to produce the n-best lists for re-ranking.

For the BTEC task, a general approach that used lattice-based system combination of two systems, a standard phrase-based system and a hierarchical phrase-based system, was taken. We also tried to process some unknown words by replacing them with the same words but different inflections that are known to the system.

1. Introduction

In the IWSLT 2010 evaluation campaign, the NICT team participated in the DIALOG translation (Chinese-English) and the BTEC (French-English) translation shared-tasks. This paper describes the machine translation approach adopted for this campaign.

For the DIALOG task, the challenge is to apply context information during translation. Furthermore, we have to deal with acoustic speech recognition (ASR) output for translation, which sometimes does not give correct recognition output. We used a state-of-the-art approach which combines the results generated from multiple translation systems to improve translation performance relative to any single system used in the combination. We employed several different approaches for system combination, including merging different phrase models, using a lattice-based system combination approach, and discriminative reranking using various global features.

For the BTEC task, we only have to deal with correct input text. Since French-English translation is one of the earliest language pairs used in machine translation research, any typical statistical machine translation system will provide a high level of performance most of the time. Therefore, we applied a lattice-based system combination approach to combine a standard phrase-based translation system and a hierarchical phrase-based translation system. Furthermore, we processed unknown words by replacing them with known words that have the same lemmas but different inflections.

The structure of the remainder of the paper is as follows: Section 2 describes each of the components that we used in our approach, Section 3 and Section 4 describe our implementation of the DIALOG translation systems and the BTEC French-English translation systems in detail and evaluate the performance of our systems, and the conclusion is given in Section 5.

2. System Components

2.1. Machine Translation Systems

We applied two machine translation models in our approach: a standard phrase-based model [1] and a hierarchical phrase-based model [2].

2.1.1. *CleopATRa*

We used a phrase-based translation system, that is similar to Pharaoh [3], a beam search decoder based on a log-linear model, CleopATRa, which is comprised of a language model, a translation model, a distortion model and word penalty. The feature weights are tuned using MERT [4].

2.1.2. *Linparse*

The hierarchical phrase-based translation system, Linparse, is similar to Hiero [5], and is based on a weighted synchronous context-free grammar (CFG) and uses a CKY algorithm with cube-pruning for efficient search. The feature functions consist of a language model, a hierarchical phrase translation model, and phrase penalty. The feature weights are also tuned using MERT [4].

2.2. Integration of Multiple Segmentation Schemes

The task of *word segmentation*, i.e., identifying word boundaries in continuous text, is one of the fundamental preprocessing steps of data-driven NLP applications like *Machine Translation* (MT). In contrast to Indo-European languages like English, many Asian languages like Chinese do not use a whitespace character to separate meaningful word units.

We use an unsupervised word segmentation algorithm that identifies word boundaries in continuous source language text in order to improve the translation quality of statistical machine translation (SMT) approaches [6].

Word segmentations that are consistent with the phrasal segmentations of SMT translation models are learned from the SMT training corpus by aligning character-wise source language sentences to word units separated by a whitespace in the target language. Successive characters aligned to the same target words are merged into a larger source language unit. Therefore, the granularity of the translation unit is defined in the given bitext context. In order to minimize the side effects of alignment errors and to achieve segmentation consistency, a Maximum-Entropy (ME) algorithm is applied to learn a source language word segmentation that is consistent with the translation model of an SMT system trained on the resegmented bitext. The process is iterated until no further improvement in translation quality is achieved.

In order to increase the coverage and to reduce the translation task complexity of the iteratively trained statistical models, our method integrates multiple segmentation schemes into the statistical translation models of a single SMT engine so that longer translation units are preferred for translation if available, and smaller translation units can be used otherwise.

The integration of multiple word segmentation schemes is carried out by merging the translation models of SMT engines trained on the characterized and iteratively learned segmentation schemes. This process is performed by linearly interpolating the model probabilities of each of the models. The advantages are twofold. Primarily it allows decoding directly from unsegmented text. Moreover, the segmentation of the source phrase can differ between models at differing iterations; removing the source segmentation at this stage makes the phrase pairs in the translation models at various stages in the iterative process consistent with one another. Consequently, duplicate bilingual phrase pairs appear in the phrase table. These duplicates are combined by summing their model probabilities prior to model interpolation.

The re-scored translation model covers all translation pairs that were learned by any of the iterative models. Therefore, the selection of longer translation units during decoding can reduce the complexity of the translation task. On the other hand, overfitting problems of single-iteration models can be avoided because multiple smaller source language translation units can be exploited to cover the given source language input parts and to generate translation hypotheses based on the concatenation of associated target phrase expressions. Moreover, the merging process increases the translation probabilities of the source/target translation parts that cover the same surface string but differ only in the segmentation of the source language phrase. Therefore, the more often such a translation pair is learned by different iterative models, the more often the respective target language expression will be exploited by the SMT decoder.

The translation of unseen data using the merged translation models is carried out by (1) characterizing the input text and (2) applying the SMT decoding in a standard way.

2.3. System Combination

A lattice-based system combination approach is applied in our model. We follow the traditional system combination approach [7, 8]. An MBR-CN framework is applied. The minimum Bayes-risk (MBR) decoder [9] is used to select the best single output to be used as the skeleton by minimizing the translation edit rate (TER) [10]. Then, the confusion network (CN) is built using the skeleton as the backbone which determines the word order of the combination. The other hypotheses are then aligned to the backbone based on the TER metric. The decoder of the CN uses only the word posterior probability, a 4-gram language model and the length penalty as the log-linear feature functions in a search process through a beam search algorithm.

2.4. SVM Reranking

2.4.1. Ranking Model Learning

Our ranking algorithm is based on a ranking approach of [11] in which we seek the maximum scored output ê from a large n-best list

$$\hat{\mathbf{e}} = \operatorname*{argmax}_{\mathbf{e} \in \mathrm{GEN}(\mathbf{f})} \mathbf{w}^{\top} \cdot \mathbf{h}(\mathbf{e}, \mathbf{f}) \quad (1)$$

where $\mathrm{GEN}(\cdot)$ is an n-best list, a set of candidate translations, generated from the input sentence $\mathbf{f}$. $\mathbf{h}(\cdot)$ defines mapping from input/output sentence pair to feature functions, and $\mathbf{w}$ is a weight vector. In training the parameter vector $\mathbf{w}$, we employed an online large-margin learning for structured output classification [12, 13, 14] based on the margin infused relaxed algorithm (MIRA) [15]. First, we generate a large n-best list $\mathbf{e}$ for m input sentences $\mathbf{f}_{1...m}$. For each iteration, we randomly choose an input sentence $\mathbf{f}_i$ and its corresponding n_i-best list $\mathbf{e}_i$. We seek a maximum scored hypothesized translation $\mathbf{e}_{ij}$ using the current weight $\mathbf{w}$

$$\mathbf{w}^{\top} \cdot \mathbf{h}(\mathbf{e}_{ij}) - \mathbf{b}(\mathbf{e}_{ij}) \quad (2)$$

where $\mathbf{h}(\mathbf{e}_{ij})$ and $\mathbf{b}(\mathbf{e}_{ij})$ are a feature vector representation and the BLEU score for $\mathbf{e}_{ij}$, respectively. Then, we update

$\mathbf{w}$ by the value of $\mathbf{w}'$ which minimizes

$$\frac{\lambda}{2}||\mathbf{w}' - \mathbf{w}||^2 + l_{ij} - \mathbf{w}'^{\top} \cdot \Delta\mathbf{h}(\mathbf{e}_{ij}) \qquad (3)$$

where l_{ij} is a loss incurred by selecting the $\mathbf{e}_{ij}$ as the best translation computed by the difference of BLEU from an oracle translation $\mathbf{e}_{i*}$

$$l_{ij} = \mathbf{b}(\mathbf{e}_{i*}) - \mathbf{b}(\mathbf{e}_{ij}) \qquad (4)$$

and $\Delta\mathbf{h}(\mathbf{e}_{ij}) = \mathbf{h}(\mathbf{e}_{i*}) - \mathbf{h}(\mathbf{e}_{ij})$. $\lambda(> 0)$ is a constant to influence the fitness to the training data. Equation 3 is solved by:

$$\mathbf{w}' = \mathbf{w}' + \min\left(\frac{l_{ij} - \mathbf{w}^{\top} \cdot \Delta\mathbf{h}_{ij}}{||\Delta\mathbf{h}_{ij}||^2}, \frac{1}{\lambda}\right) \cdot \Delta\mathbf{h}_{ij} \qquad (5)$$

Unlike the ranking SVM approach for training [16], our learning algorithm considers only a single pair of correct and incorrect translations in each iteration using the loss biased maximization in Equation 2 largely inspired by [14]. For the loss function l_{ij} and the underlying BLEU score $\mathbf{b}(\cdot)$, we applied document scaled BLEU which computes BLEU by replacing one translation $\mathbf{e}_{i1}$ with another $\mathbf{e}_{ij}$ in a set of 1-best translations $\{\mathbf{e}_{i1}\}_{i=1...m}$ [13]. Oracle translations are selected with respect to $\mathbf{b}(\cdot)$. When multiple oracle translations are found, we select the one which maximizes $\Delta\mathbf{h}(\mathbf{e}_{ij}) \cdot \mathbf{w}$ [14].

2.4.2. *Feature Functions for Re-ranking*

We used a large number of sparse binary features together with real valued features from decoders as described in [17].

Word pair features We used all possible pairs of source word and target word as our primary features. POS pairs were also extracted by replacing source words and target words with their corresponding POS tags annotated by the Stanford tagger [18]. In addition, we used simple 4-letter prefix and 4-letter suffix normalized words as the word pair features.

N-gram features In order to directly capture fluency, we extracted n-gram features in the target side from unigram to trigram. As in word pair features, n-gram features with POS/4-letter normalization were also used as our feature set.

Alignment features We used fine grained word pair features by running a word aligner which heuristically combines posterior distribution from symmetrically agreed HMM models in two directions [19]. For our heuristic combination method, we introduced ITG-constraints, instead of thresholding, by assigning zero weights to binary branching rules, and the log of posterior probabilities for bi-lexical rules. For faster Viterbi alignment computation, we employed a fast span pruning method of [20].

Syntactic features We also included syntactic features by running the Stanford parser [21] on both sides. The feature set employed in our ranking model was mainly taken from [22], namely, “Rule” and “Parent” for the rules used in the parsed tree with/without its parent category, “Word edges” for the category and span with neighboring terminal words and “NGram tree” for the minimum tree structure spanning a bigram.

Context features The DIALOG task preserves dialog context between two speakers. We directly encoded the structure as our feature set by including pairs of words between words from the current translated utterance and bags of words (BOW) from the previously “translated” last utterance from both speakers. The BOWs were collected from the n-best list of the translation. We used the BOW of the “translated” last utterance is because we prefer the bias made by the “translated” sentence, instead of the correct utterence.

2.5. Punctuation and Sentence Splitting

In all of our experiments, we trained the SMT system on punctuated data, and added punctation to the unpunctuated input. We also segmented the word sequence into a sequence of sentences that will be translated independently. The process is done in two steps using a CRF model [23] for each step. We trained both CRF models on the training data, the sentence boundaries being marked according to whether or not a sentence-final punctuation mark occurred in that position.

In the first step, the sentence boundaries are marked; in the second step, the sentence is punctuated with reference to the sentence boundaries. In both stages the CRF model assigns a label to each word. In the case of the sentence boundary model, the label indicates whether or not there is a sentence boundary after the current word. In the case of the punctuation model, the label indicates the identity of the punctuation mark to follow each word (including a label for no punctuation). The models use a feature-set typical for an n-gram tagger: n-grams of words to the left and right of the current word, and n-grams of label sequences to the left (in our experiments we considered up to 3-grams).

In addition, the punctuation model also includes features representing the first word of the current sentence unit. These features are critical in English to discriminate question sentences as question words usually occur at the start of the sentence, and are the reason we predict punctuation in two stages. For Chinese, these question words commonly occur at the end of the sentence and are therefore included in the n-gram feature set.

Table 1 shows the accuracy of our punctuation model in the number of correctly punctuated sentences. The baseline is a hidden n-gram model trained using the SRILM toolkit. The results show that the CRF model gives better accuracy in the prediction of the punctuation of the sentences for both languages.

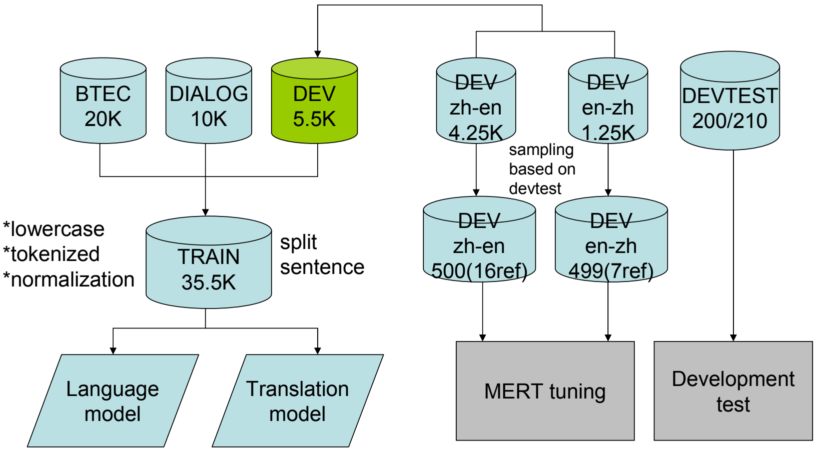

Figure 1: Data preparation for building translation model

Table 1: Punctuation Accuracy

Language	Total	Hidden	CRF
Chinese	200	126	134
English	210	74	130

3. DIALOG Task

3.1. Data Preparation

Figure 1 shows the data preparation for building our translation system. Training data was composed from both the DIALOG corpus, the BTEC corpus and the DEVSET corpus. All the data in the DEVSET for the BTEC task, using on single reference, was included for training. Only the devset for DIALOG was reserved for development testing. All of our experiment results presented in this paper are based on this testset. In total, we had around 35K sentence pairs for training.

The devset used for MERT is sampled from all of the DEVSET for BTEC. In the last year's IWSLT campaign, we introduced a devset sampling technique in which the development data were sampled from training data that are similar to the input text [24]. The similarity is measured by the BLEU using the input sentences as references. This year, we sampled from bilingual data with multiple reference translations, rather than from large amounts of DIALOG data with single reference translations, in order to avoid overfitting. We extracted 500 sentences for each translation direction. During MERT, only the training corpus for DIALOG and BTEC were used to train the translation model, but all of the data was used to build final translation model.

Some pre-processing was also carried out on the corpus before training. First, in order to avoid ambiguities when there are multiple sentences in one line (one sentence pair), we split the corpus so that one line consists of only one sentence. This is done automatically by looking at the source and the target. If they have the same number of sentences where the length ratios are quite close, we split them into multiple sentence pairs. If not, they would remain as is. After splitting, the training corpus contained around 40K sentence pairs. For the translation input text, all of the sentences are split if multiple sentences are found in one line. At the end of translation, these multiple sentences are concatenated into a single line.

We also did some normalization to the text. For English text, all the words were lowercased, any hyphens or commas were removed from between numeral words and tokenized using the standard tools provided by the Moses toolkit[1]. The Chinese word segmentation originally provided contained inconsistencies and was not usable to build the translation model. The Chinese word segmentation was therefore redone using three methods: character-based, Achilles [25] and ICTCLAS[2]. We will explain the usage of different segmentation standards in the next section. Basically, the numeral words in Chinese can be written either using Chinese characters or Arabic numbers. We converted all of the Arabic numbers to Chinese characters using a simple set of heuristics.

Our translation model was built from data containing the punctuation for both source and target languages. In the official testing, the test data is provided without punctuation to remain consistent with the format of ASR output. So, before sending the test data for translation, we restored the punctuation using the punctuation model as described in Section 2.5.

3.2. System Setup

Figure 2 shows the translation flow. We used only CleopATRa in this translation task. The language models and translation models were trained using the SRILM and Moses toolkits. First, we built different translation systems based on the different segmentation standards described earlier:

[1] http://www.statmt.org/moses/

[2] http://ictclas.org/

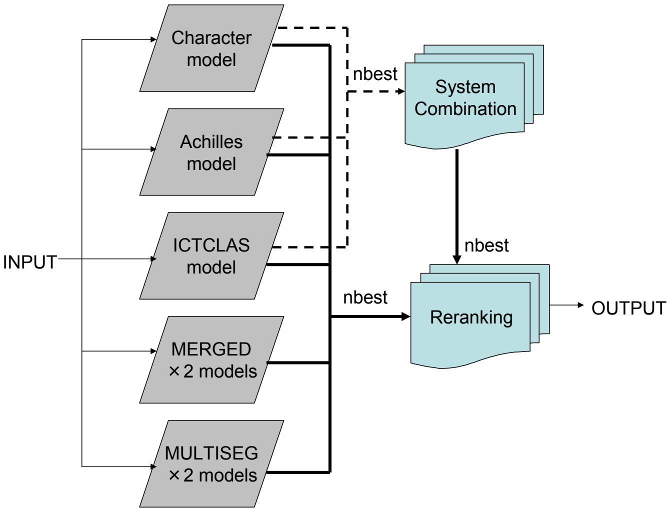

Figure 2: Translation flow

character-based, Achilles and ICTCLAS. The "Multiseg" model is a model based on the method described in Section 2.2, where different segmentation outputs are generated automatically in multiple iterative passed and joined into one phrase table. Two models are trained, using the full data (train+dev) and only the train data. The "Merged" model combines the phrase tables generated from "Multiseg" and the Achilles or Character model using linear interpolation [3].

Next, we combined the outputs from all of the systems using the system combination approach described in Section 2.3. We used only the n-best list generated from three basic systems (i.e. Character-based, Achilles and ICTCLAS models) for combination. For each model, we used both results from tuned and notuned for MERT. In total, we were combining output from 6 SMT systems. For each system, we took the 20-best outputs to produce a 120-best (1-best ASR/CRR × 20-best translation × 6 systems) list for system combination. The same translation approach was also applied to ASR 1-best output. We also built a system for ASR n-best output by generating 1-best translation for 20-best ASR output using the same 6 SMT systems for a total 120-best (20-best ASR × 1-best translation × 6 systems) list as well.

Re-ranking is trained on the development test data consisting of only 200 sentences for Chinese-English translation and 210 sentences for English-Chinese translation. 1000-best unique translations are generated from each of the three basic systems, two merged systems (merged with the Achilles model or Character model) and two multiseg systems (full-train or train-only) for the Chinese-English direction only, and also from the lattice-based system combination. All the n-best lists are casing restored and re-tokenized to meet the Penn-treebank specification in order to extract parse features. We used case-preserved BLEU for the reranking training objective. The hyperparameter λ was tuned by performing 10-fold cross validation over the development test data.

Table 2: System combination and reranking results using different segmenation standards

System	tunings	zh-en	en-zh
Character	notuned	47.49	42.72
	tuned	50.44	42.56
Achilles	notuned	46.69	42.52
	tuned	48.69	40.56
ICTCLAS	notuned	47.09	40.50
	tuned	48.88	40.55
Merged (Achilles)	notuned	46.78	-
	tuned	50.04	-
Merged (Character)	notuned	45.54	-
	tuned	48.46	-
Multiseg (full-train)	notuned	47.50	-
	tuned	50.73	-
Multiseg (train-only)	notuned	47.67	-
	tuned	49.97	-
SysComb	20-best	50.63	43.06
Rerank †	1-best	45.84	41.25
	1000-best	50.58	46.91

† Reranking results are case-preserved 10-fold averaged BLEU. 1-best translations are taken from SysComb.

Our preliminary result is shown in Table 2 for CRR output. We used Moses multi BLEU scoring for the evaluation. All the translation outputs, except re-ranking, are with punctuation: character-based for Chinese text and lowercase for English text.

The results show that the devset sampling method used for choosing the development set for MERT gives good performance for Chinese-English translation but not for English-Chinese translation. This may be because the English-Chinese devset is small, only 1.25K, so we do not have enough data to select a similar set, whereas the Chinese-English devset contains around 4.25K sentences, and therefore has a better selection. Both of the merged and multiseg systems perform better than single model systems in general. The system combination shows a slight improvement for both Chinese-English and English-Chinese translation. The re-ranking is evaluated using the 10-fold cross validation on the devtest set. Although it cannot be directly compared with other models, it shows a 4-5 BLEU point improvement.

3.3. Post-processing

Since we changed the English text to lowercase for building the translation model, we needed a model to restore the casing for the translation output. The Moses toolkit was used for building a recasing model. The training data used was from all of the DIALOG, BTEC and DEVSET data. In addition, the English translation output was also detokenized to

[3] Our preliminery results showed that merging with the Achilles model gives better translation output than the Character-based and ICTCLAS models.

the orthography of the original corpus.

A few corrections were also done for numeral words in post-processing. In pre-processing, we removed the hyphens and commas for numeral words. These hyphens and commas are therefore restored in post-processing using heuristics.

In the DIALOG translation, many numbers are involved, such as credit card number, telephone numbers, room numbers, money and general numbers. Usually, an SMT system will have difficulty preserving the order of numbers for sequential types like credit card and telephone numbers. The order will probably be arranged according to the language model. The chances of getting the order correct are small, except in cases where the number is present in the training data. So, we have made modifications to the post-processing to reorder the sequence of numbers if they are in the wrong sequence, but no changes are made in the case of a translation error.

Furthermore, there is a problem in translating Chinese general numbers that involve “万” (ten thousand). Again, most of the time, the translation will be wrong if the exact numeric sequence has not occurred in the corpus. For example, “二万五千” should be “twenty five thousand” but most likely it would become “two thousand five thousand”. So, we again carry out some pre-processing to change the Chinese number into an English-like format. In this example, it would become “二十五千” for input. Similarly, if some translations into Chinese are found in this format, we change them back to the correct format.

3.4. Official Results

Table 3 shows the official results of our DIALOG translation systems. We submitted system combination output for the IWSLT10 testset and re-ranking output for the IWSLT09 testset. This is because when we ran the test and compared the translation output, we realized that re-ranking does not give the expected results. There was a lot of missing information or erroneous translations in the translated text of the IWSLT10 testset. We think the main reason for this is that re-ranking is very sensitive to the domain of the training data. In terms of the perplexity against DIALOG training data and the average sentence length, we found that the IWSLT09 testset is close to the devtest set, which is the training data for re-ranking, but IWSLT10 is a bit out of the domain. For this reason, we believe the re-ranking was unable to give good results on the IWSLT10 testset. Since the system combination approach was not tuned to any development set, the results are more neutral. In most cases, system combination is better than re-ranking and also better than other single models. As for the ASR output, the system combination using 20-best ASR output is sometimes, but not necessarily, better, and it highly depends on the ASR performance. If the 1-best ASR output is good, then it is not necessary to use the 20-best output.

As a post evaluation, we tried to run the lattice-based system combination by also including the merged and multiseg system output for CRR track. For the IWSLT10 testset, we were unable to get any improvement (0.2272 BLEU point, -0.0060), and only slight improvement could be obtained for the IWSLT09 testset (0.3314 BLEU point, +0.0050). This means that by adding more system output to the lattice-based system combination approach does not guarantee a better translation.

4. BTEC Task

4.1. Data Preparation

For the BTEC task, we only participated in the French-English translation shared task. The data preparation for the BTEC task is straightforward. First, both the French and English texts were lowercased, tokenized and split into short sentences if multiple sentences are found in one line. Then, all hyphens were removed from the French text but only hyphens in numeral words were removed from the English text. Special treatment was given to French text by removing the character “t” for words with the pattern “*-t-il”, as this character in general does not carry any meaning [4].

The training corpus originally contained 19,972 sentence pairs, and after sentence splitting it contained 23,578 sentence pairs. The development devset1 was used for weight tuning, and devset2 and devset3 were used for development testing. As for the translation input text, all the sentences are split if multiple sentences are found in one line. At the end of translation, these multiple sentences are joined into one line sequentially.

4.2. System Setup

We used CleopATRa and Linparse to build two machine translation systems. The word alignment was done using generative models (Model 1 and HMM) with forced alignment agreement training between two directions [19], as in the word alignment features used in the re-ranking model in Section 2.4.2. The combined word alignment was annotated similarly, but using different criteria: We first ran Viterbi aligners in two directions. Then, ITG-constrained word alignment was generated by zero weighted binary branching rules and the fixed weighted bi-lexical rules of $log(1.0)$ for intersection, $log(0.5)$ for union and $log(0.001)$ for deletion. Finally, system combination was done using 20-best output from both CleopATRa and Linparse using the approach described in Section 2.3.

4.3. Unknown Word Processing

While French is a morphological rich language, English is not. French conjugation is very complicated and generates many unknown word forms for translation. Furthermore,

[4] The same pre-processing can also be done for “on” (we) and “elle” (she), but we missed them out during the data preparation. Since there are not as many of these as “il” (he/it), we assume that the results may not be too different.

Table 3: Official results for IWSLT campaign DIALOG task

System	IWSLT10 testset				IWSLT09 testset			
	English-Chinese		Chinese-English		English-Chinese		Chinese-English	
	ASR	CRR	ASR	CRR	ASR	CRR	ASR	CRR
Character	0.2743	0.3085	0.1850	0.1977	0.3273	0.3748	0.2887	0.3155
Merged (Achilles)	-	-	0.2013	0.2181	-	-	0.2875	0.3110
Multiseg (full-train)	-	-	0.1835	0.2010	-	-	0.2918	0.3066
SysComb	0.2874	**0.3172**	0.2099	**0.2332**	0.3489	0.3998	**0.3017**	**0.3264**
SysComb-20best	**0.2936**	-	**0.2103**	-	0.3462	-	0.2995	-
Rerank	0.2737	0.2999	0.1459	0.1616	**0.3679**	**0.4263**	0.2687	0.2924

there are different word forms for singular, plural, masculine and feminine. Our approach to these unknown word translations is that if a word form is unknown to the translation system, then we replace it with a word form that is known to the system. The new word form must have the same lemma as the unknown word form. We use TreeTagger[5] to get the lemmas of words.

For example, the sentence below has an unknown word ”prendront” (third-person plural form). The translation cannot be done using our system.

- combien de temps prendront les modifications ?
- how long _OOV_prendront the alterations ?

However, after we replace it with a known word “prendrons” (first-person plural form), that has the same lemma as “prendre” (to take), we can translate the sentence more correctly, though not perfectly:

- combien de temps prendrons les modifications ?
- how long does it take the alterations ?

If multiple candidates are found, we choose a known word form that has the smallest edit distance to the unknown word form. Of course, not all cases can be translated correctly by replacing the unknown word form. Sometimes, plural word forms are translated as singular word forms, or the present tense is translated into the past tense, etc. However, our experiment results showed that some improvements can be obtained, as shown in Table 4. This method can only be applied to words that generate the same lemmas but not to the derivative words, like ”baigner” (to bath) to ”baignoire” (bathtub). This method is only applicable to the translation from a morphological rich language to a morphological poor language, and not the other way round.

4.4. Post-processing

For post-processing, a recaser trained using the Moses toolkit was used for restoring the casing, and the text was detokenized. The unknown words were also kept in the target as some of these words may be proper nouns that do not need to be translated.

Table 4: Translation results before and after unknown word processing for the Linparse system

	devset2		devset3	
	before	after	before	after
# of unkwords	66	39	83	52
BLEU	0.6705	0.6747	0.6854	0.6888

Table 5: Official results for IWSLT campaign BTEC French-English translation task

System	IWSLT10 testset	IWSLT09 testset
CleopATRa	0.5146	0.5933
Linparse	0.5294	0.6108
SysComb	**0.5395**	**0.6158**

4.5. Official Results

Table 5 shows the official results for the French-English translation shared task. As predicted, system combination is better than both single systems, CleopATRa and Linparse.

5. Conclusion

In conclusion, we have successfully applied advanced techniques in statistical machine translation system in order to improve the quality of our translations. Although our re-ranking model did not give satisfactory results on the IWSLT10 testset, there is considerable room for improvement both from using more appropriate training data and by making the algorithm less sensitive to the characteristics of the development data used to train it. Although not something new, lattice-based system combination has helped to improve translation results. Joining the phrase tables generated from multiple segmentation schemes shows potential for giving better translations than only using a single segmentation scheme. It can currently only be applied to the source side, which is Chinese, but it may be used in the target side in the future. Our unknown word replacing approach in French-English translation is a good start, but not the best

[5] `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`

solution. It would be better if we could preserve word forms change in the target side as well. In that case, not only would the translation quality improve, but also translation could be made possible from a morphological poor language to a morphological rich language.

6. References

[1] P. Koehn, F. J. Och, and D. Marcu, "Statistical Phrase-Based Translation," in *Proceedings of HLT/NAACL*, 2003, pp. 48–54.

[2] D. Chiang, "A hierarchical phrase-based model for statistical machine translation," in *Proceedings of ACL*, 2005, pp. 263–270.

[3] P. Koehn, "Pharaoh: a beam search decoder for phrase-based statistical machine translation models," in *Proceedings of AMTA*, 2004, pp. 115–124.

[4] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proceedings of ACL*, 2003, pp. 160–167.

[5] D. Chiang, "Hierarchical phrase-based translation," *Computational Linguistics*, vol. 33, no. 2, pp. 201–228, 2007.

[6] M. Paul, A. Finch, and E. Sumita, "Integration of multiple bilingually-learned segmentation schemes into statistical machine translation," in *Proceedings of the Joint Fifth Workshop on SMT and MetricsMATR*, 2010, pp. 400–408.

[7] K. C. Sim, W. J. Byrne, M. J. F. Gales, H. Sahbi, and P. C. Woodland, "Consensus Network Decoding for Statistical Machine Translation System Combination," in *Proceedinsg of the ICASSP*, 2007.

[8] A.-V. I. Rosti, S. Matsoukas, and R. Schwartz, "Improved Word-level System Combination for Machine Translation," in *Proceedings of the ACL*, 2007, pp. 312–319.

[9] S. Kumar and W. Byrne, "Minimum Bayes-Risk Decoding for Statistical Machine Translation," in *Proceedings of the HLT-NAACL*, 2004, pp. 169–176.

[10] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, "A study of translation edit rate with targeted human annotation," in *Proceedings of Association for Machine Translation in the Americas*, 2006.

[11] M. Collins and N. Duffy, "New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron," in *Proceedings of ACL*, 2002, pp. 263–270.

[12] R. McDonald, K. Crammer, and F. Pereira, "Online large-margin training of dependency parsers," in *Proceedings of ACL*, 2005, pp. 91–98.

[13] T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki, "Online large-margin training for statistical machine translation," in *Proceedings of EMNLP-CoNLL*, 2007, pp. 764–773.

[14] D. Chiang, Y. Marton, and P. Resnik, "Online large-margin training of syntactic and structural translation features," in *Proceedings of EMNLP*, 2008, pp. 224–233.

[15] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *Journal of Machine Learning Research*, vol. 7, pp. 551–585, March 2006.

[16] T. Joachims, "Optimizing search engines using clickthrough data," in *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 133–142.

[17] K. Sudoh, T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki, "NTT statistical machine translation system for IWSLT 2008," in *Proceedings of the IWSLT*, 2008, pp. 92–97.

[18] K. Toutanova, D. Klein, C. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of HLT-NAACL*, 2003, pp. 252–259.

[19] P. Liang, B. Taskar, and D. Klein, "Alignment by agreement," in *Proceedings of the HLT-NAACL*, 2006, pp. 104–111.

[20] H. Zhang, C. Quirk, R. C. Moore, and D. Gildea, "Bayesian learning of non-compositional phrases with synchronous parsing," in *Proceedings of ACL-HLT*, 2008, pp. 97–105.

[21] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proceedings of ACL*, 2003, pp. 423–430.

[22] L. Huang, "Forest reranking: Discriminative parsing with non-local features," in *Proceedings of ACL-HLT*, 2008, pp. 586–594.

[23] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of ICML*, 2001, pp. 282–289.

[24] M. Utiyama, H. Yamamoto, and E. Sumita, "Two methods for stabilizing MERT: NICT at IWSLT 2009," in *Proceedings of IWSLT*, 2009, pp. 79–82.

[25] R. Zhang and E. Sumita, "Achilles: NICT/ATR Chinese Morphological Analyzer for the Fourth Sighan Bakeoff," in *Proceedings of the 6th SIGHAN workshop on Chinese Language Processing*, 2008, pp. 178–182.