Larger Feature Set Approach for Machine Translation in IWSLT 2007

Taro Watanabe, Jun Suzuki, Katsuhito Sudoh,
Hajime Tsukada, Hideki Isozaki

NTT Communication Science Laboratories
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237 Japan
{taro,jun,sudoh,tsukada,isozaki}@cslab.kecl.ntt.co.jp

Abstract

The NTT Statistical Machine Translation System employs a large number of feature functions. First, k-best translation candidates are generated by an efficient decoding method of hierarchical phrase-based translation. Second, the k-best translations are reranked. In both steps, sparse binary features — of the order of millions — are integrated during the search. This paper gives the details of the two steps and shows the results for the Evaluation campaign of the International Workshop on Spoken Language Translation (IWSLT) 2007.

1. Introduction

This paper presents NTT Statistical Machine Translation System evaluated in the evaluation campaign of International Workshop on Spoken Language Translation (IWSLT) 2007. Our system is composed of two steps: First, k-best translation candidates are generated using an efficient decoder for hierarchical phrase-based translation [1]. Next, the large k-best translation is reordered using a reranking voted perceptron [2]. Both systems employ a large number of sparse features — of the order of millions — to achieve a state of the art performance [3].

The large number of parameters are trained using an efficient online training algorithm: The decoder employs an online large-margin training method [4] that has been successfully applied in dependency parsing [5] or joint labeling/chunking [6]. The reranker uses a reranking voted perceptron which gave significant improvement in the last year's IWSLT 2006 evaluation [2]. Both systems are tuned using approximated BLEU as an objective function that scales the sentence-wise BLEU to a document-wise BLEU. Domain mismatch is handled by a simple task adaptation scheme by selecting training data that resembles a test set [7]. In order to handle the ASR's error prone input, we decoded all the n-best translations and let the reranker choose the right translation by treating the individually translated list as a single k-best list combined with the ASR's n-best list's confidence measures.

This paper is organized as follows: The overview of our decoder is presented in Section 2. We will describe the feature functions experimented in [3] together with additional features. The reranking system is described in Section 3. The reranker is biased to use a slightly different feature set to avoid over training. Both systems share the same online training algorithm, but differ in that the decoder's parameters are updated based on the dynamically generated candidate list, whereby the reranking training is based on a fixed translation candidate list. Section 4 presents the results for the evaluation campaign of IWSLT 2007.

2. Machine Translation System

We use a linear feature combination approach [8] in which a foreign language sentence f is translated into another language, for example English, e, by seeking a maximum solution:

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

where $\mathbf{h}(f, e)$ is a large-dimension feature vector. $\mathbf{w}$ is a weight vector that scales the contribution from each feature. Each feature can take any real value, such as the log of the n-gram language model to represent fluency, or a lexicon model to capture the word or phrase-wise correspondence. Under this maximization scenario, our system composed of two steps: The first step is a decoder that can efficiently generate k-best list of candidate translations in a left-to-right manner [1] based on the hierarchical phrase-based translation[9]. The second step rerank the k-best list using a reranking voted perceptron[2].

2.1. Hierarchical Phrase-based Translation

We use the hierarchical phrase-based translation approach, in which non-terminals are embedded in each phrase [9]. A translation is generated by hierarchically combining phrases using the non-terminals. Such a quasi-syntactic structure can naturally capture the reordering of phrases that is not directly modeled by a conventional phrase-based approach [10]. The non-terminal embedded phrases are learned from a bilingual corpus without a linguistically motivated syntactic structure.

Based on hierarchical phrase-based modeling, we adopted the left-to-right target generation method [1] which performed better than a phrase-based system in the last year's evaluation[2]. This method is able to generate translations efficiently, first, by simplifying the grammar so that the target side takes a phrase-prefixed form, namely a target normalized form:

$$X \rightarrow \langle \gamma, \bar{b}\beta, \sim \rangle \tag{2}$$

where X is a non-terminal, γ is a source side string of arbitrary terminals and/or non-terminals. $\bar{b}\beta$ is a corresponding target side where $\bar{b}$ is a string of terminals, or a phrase, and β is a (possibly empty) string of non-terminals. $\sim$ defines one-to-one mapping between non-terminals in γ and β.

Second, a translation is generated in a left-to-right manner, similar to phrase-based decoding using Earley-style top-down parsing on the source side [11, 1, 12]. The basic idea is to perform top-down parsing so that the projected target side is generated in a left-to-right manner. The search is guided with a push-down automaton, which keeps track of the span of uncovered source word positions. Combined with the rest-cost estimation aggregated in a bottom-up way, our decoder efficiently searches for the most likely translation.

The use of a target normalized form further simplifies the decoding procedure, at the expense for expressiveness. Since the rule form does not allow any holes in the target side, the integration with an n-gram language model is straightforward: the prefixed phrases are simply concatenated and intersected with an n-gram.

2.2. Features

2.2.1. Baseline Features

The hierarchical phrase-based translation system employs standard real valued value features:

- n-gram language model to capture the fluency of the target side.
- Hierarchical phrase translation probabilities in both directions, $h(\gamma|\bar{b}\beta)$ and $h(\bar{b}\beta|\gamma)$, estimated by relative counts, $\text{count}(\gamma, \bar{b}\beta)$ [9].
- Word-based lexically weighted models of $h_{lex}(\gamma|\bar{b}\beta)$ and $h_{lex}(\bar{b}\beta|\gamma)$ using lexical translation models[9].
- Word-based insertion/deletion penalties that penalize through the low probabilities of the lexical translation models [13].
- Word/hierarchical-phrase length penalties.
- Backtrack-based penalties inspired by the distortion penalties in phrase-based modeling [1].

2.2.2. Sparse Features

In addition to the baseline features, a large number of binary features are integrated in our MT system [3]. The features are designed with decoding efficiency in mind and are based on the word alignment structure preserved in hierarchical phrase translation pairs [14]. When hierarchical phrases are extracted, the word alignment is preserved. If multiple word alignments are observed with the same source and target sides, only the most frequently observed word alignment is kept to reduce the grammar size.

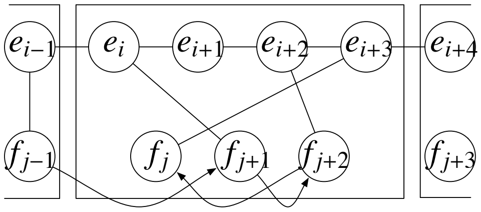

Figure 1: An example of sparse features for a phrase translation.

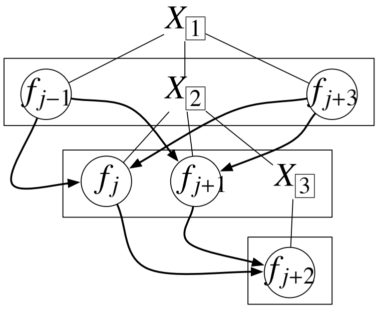

Figure 2: Example hierarchical features.

Using the word alignment structure inside hierarchical phrases, we employs following feature set.

- *Word pair features* directly capture the source/target word correspondence represented by the word alignment, such as (e_i, f_{j+1}), (e_{i+2}, f_{j+2}) and (e_{i+3}, f_j) in Figure 1.

  In addition to the unigram word pair feature, bigram word pair features are also used to capture the contextual dependency, such as $((e_{i-1}, f_{j-1}), (e_i, f_{j+1}))$, $((e_i, f_{j+1}), (e_{i+2}, f_{j+2}))$ and $((e_{i+2}, f_{j+2}), (e_{i+3}, f_j))$ indicated by the arrows in Figure 1.

  We assume that the bigram of the word pairs will follow the target side ordering. Extracting bigram word pair features following the target side ordering implies that the corresponding source side is reordered according to the target side. The reordering of hierarchical phrases is represented by using contextually dependent word pairs across their boundaries, as with the feature $((e_{i-1}, f_{j-1}), (e_i, f_{j+1}))$.

- *Insertion/deletion features* are integrated in which no word alignment is associated in the target/source side. Inserted words are associated with all the words in the source sentence, such as $(e_{i+1}, f_1), ..., (e_{i+1}, f_J)$ for the non-aligned word e_{i+1} with the source sentence f_1^J in Figure 1. In the same way, we use hierarchical phrase-wise deletion features by associating each inserted source word in a phrase to all the target words in the same phrase.

- *Target bigram features* are also included to directly capture the fluency as in the n-gram language model [15], such as $(e_{i-1}, e_i), (e_i, e_{i+1}), (e_{i+1}, e_{i+2})$... in Figure 1.

- *Hierarchical features* capture dependencies the source words in a parent phrase to the source words in child phrases, such as (f_{j-1}, f_j), (f_{j-1}, f_{j+1}), (f_{j+3}, f_j), (f_{j+3}, f_{j+1}), (f_j, f_{j+2}) and (f_{j+1}, f_{j+2}) as indicated by the arrows in Figure 2. The hierarchical features are extracted only for those source words that are aligned with the target side to limit the feature size.

In order to achieve the generalization capability, we introduce normalized tokens for each surface form [3].

- Word class/part-of-speech/named entity. Words are clustered by mkcls [16]. The part-of-speech (POS) and named entity (NE) tags are also integrated to capture linguistic characteristics when taggers are available.

  A unique word class is assigned to each surface form. However, multiple POS/NE are potentially assigned to each surface word. In our approach, we do not disambiguate labels, but simply collect a surface word to multiple tags dictionary. Those tags are integrated by first running a tagger on the training data. Then, a surface form to POS/NE dictionary is generated by collecting all possible tags for each word.

- Synsets from WordNet. In order to represent semantic correspondence, we introduced synset categories for the English side. The synset mapping is potentially one-to-many as with the POS/NE features.

- 4-letter prefix and suffix. For instance, the word "violate" is normalized to "viol+" and "+late" by taking the prefix and suffix, respectively.

- Digits replaced by a sequence of "@". For example, the word "2007/6/27" is represented as "@@@@/@/@@". Since all the numerals are spell-out, this feature is applicable only to Chinese and Japanese.

We consider all possible combination of those token types. For example, an English/Arabic word pair feature (violate, tnthk) is normalized and expanded to (viol+, tnthk), (viol+, tnth+), (violate, tnth+), etc. using the 4-letter prefix token type. As discussed above, the POS/NE/synsets labels are assigned by a one-to-many dictionary. Then, each surface form is expanded to all possible labels, then, all possible features are extracted.

Algorithm 1 Online Training Algorithm for decoder

```
   Training data: $\mathcal{T} = \{(f^t, \mathbf{e}^t)\}_{t=1}^{T}$
   $m$-best oracles: $\mathcal{O} = \{\}_{t=1}^{T}$
   $i = 0$
1: for $n = 1, ..., N$ do
2:     for $t = 1, ..., T$ do
3:         $\mathcal{C}^t \leftarrow \mathrm{best}_k(f^t; \mathbf{w}^i)$
4:         $\mathcal{O}^t \leftarrow \mathrm{oracle}_m(\mathcal{O}^t \cup \mathcal{C}^t; \mathbf{e}^t)$
5:         $\mathbf{w}^{i+1}$ = update $\mathbf{w}^i$ using $\mathcal{C}^t$ w.r.t. $\mathcal{O}^t$
6:         $i = i + 1$
7:     end for
8: end for
9: return $\frac{\sum_{i=1}^{NT} \mathbf{w}^i}{NT}$
```

2.3. Training

Algorithm 1 is our generic online training algorithm. The algorithm is slightly different from other online training algorithms [17, 18] in that we keep and update oracle translations, which is a set of good translations reachable by a decoder according to a metric, e.g. BLEU [19]. In line 3, a k-best list is generated by $\mathrm{best}_k(\cdot)$ using the current weight vector $\mathbf{w}^i$ for the training instance of $(f^t, \mathbf{e}^t)$. Each training instance has multiple (or, possibly one) reference translations $\mathbf{e}^t$ for the source sentence f^t. Using the k-best list, m-best oracle translations $\mathcal{O}^t$ are updated by $\mathrm{oracle}_m(\cdot)$ for every iteration (line 4). Usually, a decoder cannot generate translations that exactly match the reference translations due to its beam search pruning and OOV. Thus, we cannot always assign scores to each reference translation. Therefore, possible oracle translations are maintained according to an objective function. The problem can be resolved by approximately precomputing the oracle translations in advance [17]. Liang et at. [18] presented a similar updating strategy in which parameters were updated toward an oracle translation found in $\mathcal{C}^t$, but ignored potentially better translations discovered in the past iterations.

A new $\mathbf{w}^{i+1}$ is computed using the k-best list $\mathcal{C}^t$ with respect to the oracle translations $\mathcal{O}^t$ (line 5). After N iterations, the algorithm returns an averaged weight vector to avoid overfitting (line 9).

When updating parameters in line 5, we use the Margin Infused Relaxed Algorithm (MIRA) [4] which is an online version of the large-margin training algorithm for structured classification [20] that has been successfully used for dependency parsing [5] and joint-labeling/chunking [6]. Line 5 of the weight vector update procedure in Algorithm 1 is replaced by the solution of:

$$\begin{aligned} &\hat{\mathbf{w}}^{i+1} = \operatorname*{argmin}_{\mathbf{w}^{i+1}} \frac{1}{2}||\mathbf{w}^{i+1} - \mathbf{w}^{i}||^2 + C \sum_{\hat{e},e'} \xi(\hat{e}, e') \\ &\text{subject to} \\ &s^{i+1}(f^t, \hat{e}) - s^{i+1}(f^t, e') + \xi(\hat{e}, e') \geq L(\hat{e}, e'; \mathbf{e}^t) \\ &\xi(\hat{e}, e') \geq 0 \\ &\forall \hat{e} \in \mathcal{O}^t, \forall e' \in \mathcal{C}^t \end{aligned} \tag{3}$$

where $s^i(f^t, e) = \left\{\mathbf{w}^i\right\}^\top \cdot \mathbf{h}(f^t, e)$. $\xi(\cdot)$ is a non-negative slack variable and $C \geq 0$ is a constant to control the influence to the objective function. A larger C implies larger updates to the weight vector. $L(\cdot)$ is a loss function, for instance difference of BLEU, that measures the difference between $\hat{e}$ and e' according to the reference translations $\mathbf{e}^t$.

In this update, a margin is created for each correct and incorrect translation at least as large as the loss of the incorrect translation. A larger error means a larger distance between the scores of the correct and incorrect translations. Only active features constrained by Eq. 3 are kept and updated, unlike offline training in which all possible features have to be extracted and selected in advance.

2.4. Approximated BLEU

We used the BLEU score [19] as the loss function computed by:

$$\mathrm{BLEU}(E; \mathbf{E}) = \exp\left(\frac{1}{N}\sum_{n=1}^{N} \log p_n(E, \mathbf{E})\right) \cdot \mathrm{BP}(E, \mathbf{E}) \tag{4}$$

where $p_n(\cdot)$ is the n-gram precision of hypothesized translations $E = \{e^t\}_{t=1}^T$ given reference translations $\mathbf{E} = \{\mathbf{e}^t\}_{t=1}^T$ and $\mathrm{BP}(\cdot) \leq 1$ is a brevity penalty. BLEU is computed for a set of sentences, not for a single sentence. Our algorithm requires frequent updates on the weight vector, which implies higher cost in computing the document-wise BLEU. [17] and [18] solved the problem by introducing a sentence-wise BLEU. However, the use of the sentence-wise scoring does not translate directly into the document-wise score because the n-gram precision statistics and the brevity penalty statistics are aggregated for a sentence set. Thus, we use an approximated BLEU score that basically computes BLEU for a sentence set, but accumulates the difference for a particular sentence [2].

The approximated BLEU is computed as follows: Given oracle translations $\mathcal{O}$ for $\mathcal{T}$, we maintain the best oracle translations $O_1^T = \left\{\hat{e}^1, ..., \hat{e}^T\right\}$ that is treated as a "bed" document. The approximated BLEU for a hypothesized translation e' for the training instance $(f^t, \mathbf{e}^t)$ is computed over the bed O_1^T except for $\hat{e}^t$, which is replaced by e':

$$\mathrm{BLEU}(\{\hat{e}^1, ..., \hat{e}^{t-1}, e', \hat{e}^{t+1}, ..., \hat{e}^T\}; \mathbf{E})$$

The loss computed by the approximated BLEU measures the document-wise loss of substituting the correct translation $\hat{e}^t$ into an incorrect translation e'. The score can be regarded as a normalization which scales a sentence-wise score into a document-wise score.

Algorithm 2 Online Training Algorithm for Reranker

```
Training data: $\mathcal{T} = \{(f^t, \mathcal{C}^t, \mathbf{e}^t)\}_{t=1}^T$
for $n = 1, ..., N$ do
    $\mathbf{w}^n = \mathbf{w}^{n-1}$
    for $t = 1, ..., T$ do
        $\mathcal{R} = \mathrm{rerank}(\mathcal{C}^t; \mathbf{w}^n)$
        for $i = 1, ..., |\mathcal{R}|$ do
            for $j = i + 1, ..., |\mathcal{R}|$ do
                if $L(\mathcal{R}_j, \mathcal{R}_i; \mathbf{e}^t) > 0$ then
                    $\mathbf{w}^n$ = update $\mathbf{w}^n$ using $\mathcal{R}_i$ and $\mathcal{R}_j$
                end if
            end for
        end for
    end for
end for
return $\{\mathbf{w}^n\}_{n=1}^N$
```

3. Reranking System

Our reranking system is basically identical to the system presented in the last year's IWSLT 2006 evaluation [2] that is based on the parse reranking method explained in [21]. We first generate n-best lists of candidate translations from the decoder, then train reranking model using the development set with additional features by ranking voted perceptron. Finally, during the testing, we rerank the k-best list of test data from the decoder by the parameters for the reranking. A separately trained reranking model is used for the ASR's n-best list. The reranker selects the best translation out of the merged $k \cdot n$-best list generated by translating all the sentences in the n-best list.

3.1. Features

The reranking system employs a slightly different feature set from the baseline decoder. First, we use all the baseline features from the decoder. The decoder's sparse feature parameters are treated as a single feature function. Second, we include only unigram and bigram of word pair features in Section 2.2.2 to avoid over training. The word pairs are extracted by separately running IBM Model 1 in both directions, not directly from the word alignment annotation preserved in the hierarchical phrases from the decoder. The surface form is factored using English POS only, but used different algorithm to perform POS tagging to achieve different types of generalization. We also include various confidence measures available from the ASR's n-best and lattice outputs when reranking ASR $k \cdot n$ translations.

3.2. Training

Table 1: Bilingual data

	Arabic-to-English		Chinese-to-English		Italian-to-English		Japanese-to-English	
# sentences	832,912		3,268,916		854,871		1,055,144	
# words	21,171,984	25,062,213	51,938,862	57,289,887	24,035,970	24,041,843	10,811,003	8,646,894
vocabulary	290,826	132,915	824,720	961,193	117,914	67,262	384,236	254,442
other sources	LDC(news)		LDC(news,lexicon)		EuroParl		NiCT, others	

Algorithm 3 Decoding Algorithm for Reranker

```
$k$-best translation list: $(f, \mathcal{C})$
Weight vectors: $\{\mathbf{w}^n\}_{n=1}^N$
Votes: $\mathcal{V} = \mathbf{0}$
for $n = 1, ..., N$ do
    $\hat{i} = \mathrm{argmax}_i \{\mathbf{w}^n\}^\top \cdot \mathbf{h}(f, \mathcal{C}_i)$
    $\mathcal{V}_{\hat{i}} = \mathcal{V}_{\hat{i}} + 1$
end for
return $\mathcal{C}_{\hat{i}}$ where $\hat{i} = \mathrm{argmax}_i \mathcal{V}_i$
```

The training algorithm is presented in Algorithm 2. The major difference from Alg. 1 is that the training data comes from a static k-best list candidates $\mathcal{C}^t$ from the decoder. In line 4, $\mathcal{C}^t$ is reranked by $\mathrm{rerank}(\cdot)$ using the current weight vector $\mathbf{w}^n$ for the training instance $(f^t, \mathcal{C}^t, \mathbf{e}^t)$. Each translation candidate in the reranked k-best list $\mathcal{R}$ is pair wise compared in line 5 and 6. The weight vector is updated when incorrect ranking is found between $\mathcal{R}_i$ and $\mathcal{R}_j$ indicated by a loss function $L(\cdot)$ (line 7 and 8). After N iterations, the algorithm returns N weight vectors $\{\mathbf{w}^n\}_{n=1}^N$. When testing, the best hypothesized translation out of $\mathcal{C}$ is selected by voting as in Algorithm 3.

The weight vector update procedure in line 8 is based on an perceptron algorithm with the update amount scaled by the loss function $L(\cdot)$.

$$\mathbf{w}^n = \mathbf{w}^n + L(\mathcal{R}_j, \mathcal{R}_i; \mathbf{e}^t) \cdot \left(\mathbf{h}(f^t, \mathcal{R}_j) - \mathbf{h}(f^t, \mathcal{R}_i)\right) \quad (5)$$

As our loss function, we employed the difference of the approximated BLEU in Section 2.4, but used a set of 1-best translations from the decoder as our bed document, instead of oracle translations. The idea is to directly measure the gain or loss by selecting the translation different from the original 1-best translation of the decoder.

4. Evaluation

4.1. Data

The major training data comes from IWSLT supplied data, a subset of BTEC[22]. We also used common bilingual data either in the public domain or from the LDC as indicated in Table 1. Additional data for Arabic/English and Chinese/English comes from a set of LDC bilingual news data, lexicon and the named entity list. For Italian/English, a portion of EuroParl [23] was extracted. Additional data for Japanese/English come from the news data and miscellaneous text data supplied by NiCT [24], together with textbook-like data, a lexicon and a named entity list in the public domain [1]. The corpus statistics is presented in Table 1. Since there exists larger mismatch with the IWSLT condition, we extracted texts that do not contain any digits by discarding sentences that match the regular expression, "[0-9]". We used a development set of 4, 5 and 5b for estimating parameters both of the decoder and the reranker, since those data include ASR's outputs.

Table 2: The source language perplexity for the "clean" development and test set.

	dev set	test set
Arabic-to-English	561.96	214.99
Italian-to-English	277.24	271.39
Japanese-to-English	51.29	13.45
Chinese-to-English	188.49	73.18

Tokenization/tagging are performed by the following tools: English data is POS tagged by a MaxEnt-based tool [25] for use in the decoder, and by a rule-based Brill's POS tagger for reranking. Arabic data is tokenized by simply isolating Arabic scripts. Italian data is POS tagged by tree-tagger [26]. Japanese/Chinese texts are POS tagged/NE chunked [27]. After tokenization, we removed all the punctuation marks in the source side of bilingual data and lowercased the texts. The English side of the bilingual data is case/punctuation preserved.

4.2. Task Adaptation

As discussed in Section 4.1, we extracted bilingual data from various sources, ranging from in-domain travel related data to out-of-domain news, miscellaneous texts and lexicons. Their characteristics are very different from the style in the IWSLT development and test conditions. Table 2 shows the development/test set perplexity of the source side language computed by the trigram of the source part of the IWSLT's supplied bilingual texts. Even the development and test data is different from the IWSLT's training data.

Therefore, we performed a simple task adaptation scheme based on [7]. For each sentence in a test/development set, we sampled 100 sentences from all the bilingual training data using the source sentence's ngram precision metric. Thus, the task adapted training data will amount to

[1] `http://www.csse.monash.edu.au/~jwb/japanese.html`

Table 3: Evaluation results for IWSLT 2007. The primary submissions are indicated by †.

		ar-en		it-en		ja-en		zh-en	
		BLEU	NIST	BLEU	NIST	BLEU	NIST	BLEU	NIST
		[%]		[%]		[%]		[%]	
ASR	ASR-1-best + 1-best	36.26 †	6.61	28.68 †	6.36	35.35 †	6.43		
	ASR-20-best + rerank (devset)	30.37	5.89	26.01	5.85	35.33	6.33		
	ASR-1-best + rerank (devset+IWSLT)	39.09	6.86	28.34	6.31	38.74	6.84		
clean	1-best	34.03 †	6.50	30.91 †	6.73	43.65 †	7.56	26.27	5.71
	rerank (devset)	34.46	6.41	29.83	6.59	44.59	7.63		
	rerank (devset+IWSLT)	36.03	6.68	30.68	6.67	45.98	7.88	27.89 †	6.04

Table 4: Post-evaluation results for IWSLT 2007.

	Arabic-to-English		Italian-to-English		Japanese-to-English		Chinese-to-English	
	BLEU	NIST	BLEU	NIST	BLEU	NIST	BLEU	NIST
	[%]		[%]		[%]		[%]	
ASR-1best	48.64	6.91	36.71	7.33	43.69	7.16		
clean-1-best	48.70	6.84	39.44	7.76	51.42	8.05	35.27	5.93

50,000 sentences for a set of 500 test sentences with duplicates. From the sampled data, we generated a hierarchical phrase translation table and a lexical translation table by first running an in-house developed HMM-based "alignment by agreement" word alignment tool [28] in two directions. Then, hierarchical phrase translation pairs were extracted [1] after the grow-diag-final word alignment heuristic [10].

The parameters are estimated using the hierarchical phrase translation table sampled for the development data. For testing, we used the same parameters, but replaced the phrase translation table and the lexicon model sampled for the test data. The ngram language models are separately estimated from the English side of the IWSLT supplied bilingual data and the sampled bilingual data. The online large-margin training for our decoder was performed 200 to 300 iterations using 1-oracle 1-best constraints. The iterations varies depending on the language pairs.

The reranker was tuned on the 1000-best outputs from the decoder. We used two different data sets. One was the same development set used for the parameter tuning for our decoder (devset). In addition to the development data, we trained the reranking model on the IWSLT supplied data consisting of 20,000 sentences together with the 1-best development data sampled from the larger training data (devset+IWSLT). We expect that the sampled training data would produce further benefits in the testing condition.

4.3. Results

Our results in BLEU [19] and NIST [29] are presented in Table 3. As discussed in Section 4.2, the closer the test data to the IWSLT supplied training data, the better the BLEU scores when the training data size for our reranker is also increased. However, the larger data did not provide significant improvements for the test data when the testing and the training conditions are different as in the Italian-to-English translation task. Although we exploited a set of confidence measures from the ASR's word lattice structure, we achieved almost no gains to a simpler 1-best translation method. One of the strange behavior was observed in the Arabic-to-English task: The ASR output translation was better than the clean input translation. We believe that the ASR acts as a normalizer for the input text, which reduces the gap between the development set and the final test set.

Contrary to our previous studies on an Arabic-to-English translation task [1], our results are considerably lower than other systems. We believe that this is mainly due to the mismatch observed between the development and the test conditions. Since our method involves a large number of sparse features, it is very sensitive to the closeness to the settings.

4.4. Post-Evaluation Results

As indicated by Table 2 and the official results in Table 3, we hypothesized that our approach was very sensitive to the style mismatch. Thus we conducted further experiments by choosing the right development set for training.

We used devset 1 and 2 for estimating parameters. The devset 3 was held-out as a development test to terminate the iterations of Algorithm 1. The hierarchical phrase translation tables were acquired only from the IWSLT supplied data together with devset 4 and 5. For the Italian-to-English task, since larger distance was observed in terms of perplexities, we employed devset 4 and 5 for the parameter estimation. The devset 5b was used as a development test. The phrase translation tables was extracted from the IWSLT supplied bilingual data mixed with devset 1 through 3 and devset 5b.

The results are summarized in Table 4. We translated ASR-1best and clean data, and employed no reranking for this set of experiments, since we observed no gains. We achieved significant improvements by simply selecting the right development set. It was also observed that longer iterations easily overfit to the development data, hence resulted in worse performance for the development test and the final test. We believe that this might be the case for the IWSLT-style data condition, but will be investigated in our future work.

5. Conclusion

We evaluated the NTT Statistical Machine Translation System for the evaluation campaign of IWSLT 2007. The system consists of two steps, decoding and reranking, by integrating large number of feature functions, e.g. syntactic features. Training data is sampled using the test set ngram precision metrics from the universe of bilingual data from various sources. The large number of parameters for the decoder is tuned to a small development set. Larger training data is employed for the tuning the reranking model. Our future work involves more features and a better smoothing method to avoid over training effects observed in this evaluation.

6. Acknowledgements

We would like to thank the IWSLT 2007 organizing committee for their efforts to coordinating the evaluation campaign. This work is partly supported by MEXT Grant-In-Aid for Scientific Research of Priority Areas: Cyber Infrastructure for the Information-Explosion Era.

7. References

[1] T. Watanabe, H. Tsukada, and H. Isozaki, "Left-to-right target generation for hierarchical phrase-based translation," in *Proc. of COLING/ACL 2006*, Sydney, Australia, July 2006, pp. 777–784.

[2] T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki, "NTT Statistical Machine Translation for IWSLT 2006," in *Proc. of IWSLT 2006*, Kyoto, Japan, 2006, pp. 95–102.

[3] ——, "Online large-margin training for statistical machine translation," in *Proc. of the EMNLP-CoNLL 2007*, pp. 764–773.

[4] 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.

[5] R. McDonald, K. Crammer, and F. Pereira, "Online large-margin training of dependency parsers," in *Proc. of ACL 2005*, Ann Arbor, Michigan, June 2005, pp. 91–98.

[6] N. Shimizu and A. Haas, "Exact decoding for jointly labeling and chunking sequences," in *Proc. of the COLING/ACL 2006 Main Conference Poster Sessions*, Sydney, Australia, July 2006, pp. 763–770.

[7] A. Ittycheriah and S. Roukos, "Direct translation model 2," in *Proc. of HLT/NAACL 2007*, Rochester, New York, April 2007, pp. 57–64.

[8] F. J. Och, "Minimum error rate training in statistical machine translation," in *Proc. of ACL 2003*, Sapporo, Japan, July 2003, pp. 160–167.

[9] D. Chiang, "A hierarchical phrase-based model for statistical machine translation," in *Proc. of ACL 2005*, Ann Arbor, Michigan, June 2005, pp. 263–270.

[10] P. Koehn, F. J. Och, and D. Marcu, "Statistical phrase-based translation," in *Proc. of NAACL 2003*, Edmonton, Canada, 2003, pp. 48–54.

[11] D. Wu and H. Wong, "Machine translation with a stochastic grammatical channel," in *Proc. of COLING 98*, Montreal, Quebec, Canada, 1998, pp. 1408–1415.

[12] A. Zollmann and A. Venugopal, "Syntax augmented machine translation via chart parsing," in *Proc. of WSMT 2006*, New York City, June 2006, pp. 138–141.

[13] O. Bender, R. Zens, E. Matusov, and H. Ney, "Alignment templates: the RWTH SMT system"," in *Proc. of IWSLT 2004*, Kyoto, Japan, 2004, pp. 79–84.

[14] R. Zens and H. Ney, "Discriminative reordering models for statistical machine translation," in *Proc. of WSMT 2006*, New York City, June 2006, pp. 55–63.

[15] B. Roark, M. Saraclar, M. Collins, and M. Johnson, "Discriminative language modeling with conditional random fields and the perceptron algorithm," in *Proc. of ACL 2004*, Barcelona, Spain, July 2004, pp. 47–54.

[16] F. J. Och, "An efficient method for determining bilingual word classes," in *Proc. of EACL 1999*, Bergen, Norway, 1999, pp. 71–76.

[17] C. Tillmann and T. Zhang, "A discriminative global training algorithm for statistical MT," in *Proc. of COLING/ACL 2006*, Sydney, Australia, July 2006, pp. 721–728.

[18] P. Liang, A. Bouchard-Côté, D. Klein, and B. Taskar, "An end-to-end discriminative approach to machine translation," in *Proc. of COLING/ACL 2006*, Sydney, Australia, July 2006, pp. 761–768.

[19] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proc. of ACL 2002*, Philadelphia, Pennsylvania, 2002, pp. 311–318.

[20] B. Taskar, D. Klein, M. Collins, D. Koller, and C. Manning, "Max-margin parsing," in *Proc. of EMNLP 2004*, Barcelona, Spain, July 2004, pp. 1–8.

[21] M. Collins and N. Duffy, “New Ranking Algorithms for Parsing and Tagging: Kernels over Discrete Structures, and the Voted Perceptron,” in *Proc. of ACL’2002*, 2002, pp. 263–270.

[22] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, “Toward a broad-coverage bilingual corpus for speech translation of travel conversation in the real world,” in *Proc. of LREC 2002*, Las Palmas, Spain, 2002.

[23] P. Koehn, “Europarl: A parallel corpus for statistical machine translatio,” in *Proc. of MT Summit X*, 2005, pp. 79–86.

[24] M. Utiyama and H. Isahara, “Reliable measures for aligning japanese-english news articles and sentences,” in *Proc. of ACL 2003*, Sapporo, Japan, 2003, pp. 72–79.

[25] Y. Tsuruoka and J. Tsujii, “Bidirectional inference with the easiest-first strategy for tagging sequence data,” in *Proc. of HLT/EMNLP 2005*, Vancouver, British Columbia, Canada, October 2005, pp. 467–474.

[26] H. Schmid, “Probabilistic part-of-speech tagging using decision trees,” in *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK, 1994, pp. 44–49.

[27] K. Saito and M. Nagata, “Multi-language named entity recognition system based on hmm,” in *Proceeding of ACL2003 Workshop on Multilingual and Mixed-Language Named Entity Recognition*, 2003, pp. 41–48.

[28] P. Liang, B. Taskar, and D. Klein, “Alignment by agreement,” in *Proc. of HLT/NAACL 2006*, New York City, USA, June 2006, pp. 104–111.

[29] G. Doddington, “Automatic evaluation of machine translation quality using n-gram co-occurrence statistics,” in *In Proc. ARPA Workshop on Human Language Technology*, 2002.