The HIT-LTRC Machine Translation System for IWSLT 2012

Xiaoning Zhu, Yiming Cui, Conghui Zhu, Tiejun Zhao, Hailong Cao

Language Technology Research Center
Harbin Institute of Technology, China
{xnzhuymcui,chzhu,tjzhao,hailong}@mtlab.hit.edu.cn

Abstract

In this paper, we describe HIT-LTRC’s participation in the
IWSLT 2012 evaluation campaign. In this year, we took part
in the Olympics Task which required the participants to
translate Chinese to English with limited data.

Our system is based on Moses\cite{1}, which is an open source
machine translation system. We mainly used the phrase-based
models to carry out our experiments, and factored-based
models were also performed in comparison. All the involved
tools are freely available.

In the evaluation campaign, we focus on data selection,
phrase extraction method comparison and phrase table
combination.

1. Introduction

This paper describes the Statistical Machine Translation (SMT)
system explored by the Language Technology Research
Center of Harbin Institute of Technology (HIT-LTRC) for
IWSLT 2012. Generally, our system was based on Moses, and
phrase-based models were used.

In Olympics shared task, the training data was limited to
the supplied data including HIT Olympic Bilingual Corpus
(HIT)\cite{2} and Basic Travel Expression Corpus (BTEC)\cite{3}.
Although the two corpora are both oral corpus, there are still
some differences between them. For example, the BTEC
corpus is travel-related, and the HIT corpus is mainly about
the Olympic Games. Besides this, the organizer of IWSLT
2012 also provided two development sets which are selected
from the HIT and BTEC corpus respectively. Because the
training data is limited by the above corpus, in order to get a
better performance, we need to excavate all the potential of
the two corpora, including the development sets.

One key problem of the SMT system is how to extract the
phrase. Giza++\cite{4} is a popular word alignment tool which can
produce word alignment information with parallel corpus. By
using heuristic phrase extraction method, we can extract phrases
with the alignment. Compared with heuristic phrase
extraction method, Pialign\cite{5} is an unsupervised model for
joint phrase alignment and extraction using nonparametric
Bayesian methods and inversion transduction grammars
(ITGs). We compared the phrase table extracted by the two
phrase extraction methods in many ways, such as the size, the
quality, and the differences of two methods.

System combination has been approved to improve
machine translation performance significantly. With several
machine translation systems’ outputs, researchers can get a
better translation by combining the outputs. But in this paper,
we didn’t combine the outputs; instead we combine the
models generated by Giza++ and Pialign. It is shown that we
can get a better performance by model combination.

2. Phrase-based System

Our primary system is based on Moses with a phrase-based
model. Under the log-linear framework\cite{6}, when given a
source sentence \( f \), we can get a translation \( e \) as follows:

\[
p(e|f;\lambda) = \frac{\exp(\lambda^T h(f,e))}{Z(\lambda)}
\]

with

\[
Z(\lambda) = \sum \exp(\lambda^T h(f,e))
\]

where \( h(f,e) \) denotes the feature vector of the pair \((f,e)\),
and \( \lambda \) is its corresponding weight vector. \( h(f,e) \) contains
14 features and they are divided into following categories:

- Bidirectional translation probabilities;
- Bidirectional lexical translation probabilities;
- MSD-reordering model;
- Distortion model;
- Language model;
- Word penalty;
- Phrase penalty.

2.1. Pre-processing

The Chinese sentences supplied by the organizer were not
segmented, so we used the Stanford Word Segmenter\cite{7}
to segment the Chinese sentences with the PKU model. The
English sentences were not tokenized, thus we used the open
source tools supplied by Moses to tokenize them. We also
lowercased all the English data for training. There are many
English punctuation characters in Chinese sentences (and vice
versa), so we wrote some scripts to change all the punctuation
characters in order.

2.2. Training

In the training step, we used Giza++ to get alignments and
combined the alignments with grow-diag-final-and method.
With the alignments, we can extract phrases with heuristic
phrase extraction method and generate the translation model.
Besides, we also used Pialign to generate the alignments and
phrases.
The language model was built with SRILM toolkit[5]. A 5-gram language model was used for decoding. The corpus we used to build the language model is all the supplied data, including training data and development data.

2.3. Decoder
The decoder used in our system is Moses.

2.4. Tuning
The parameters were tuned on the development set with standard trainer MERT[9]. When running MERT, the k-best-list-size was set as 100 and BLEU4[10] was selected as the evaluation metric.

2.5. Post-processing
The translations were post-processed after decoding.
- All the Chinese words in output were deleted. Because there are many names in the test set, and most of them can’t be translated, so we deleted them;
- The English sentences were de-tokenized;
- The English sentences were re-cased by the recaser tools provided by Moses.

3. Corpus
The IWSLT organizer provided two training corpora, including HIT corpus and BTEC corpus. HIT corpus is a multilingual oral corpus developed for the Beijing 2008 Olympic Games. There are five domains in HIT corpus, including traveling, dining, sports, traffic and business. The BTEC corpus is also an oral corpus containing tourism-related sentences. Besides the training corpus, they also provided two development corpus, which were extracted from the HIT corpus and BTEC corpus. In the following paper, we use HIT_train, HIT_dev, BTEC_train, BTEC_dev to denote four corpora respectively.

In our system, we used HIT_train, BTEC_train, BTEC_dev, HIT_dev as our training data. And HIT_dev was also used as our development set. We also random sampled 1000 sentences from HIT corpus as our test set.

The detail of the corpus is presented in Table 1.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>align</th>
<th>total</th>
<th>common</th>
<th>different</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Giza+</td>
<td>1182913</td>
<td>409443</td>
<td>773470</td>
<td>976077</td>
</tr>
<tr>
<td>2 Giza+</td>
<td>1208128</td>
<td>418788</td>
<td>789340</td>
<td>994579</td>
</tr>
<tr>
<td>3 Giza+</td>
<td>1236688</td>
<td>428377</td>
<td>808306</td>
<td>1017200</td>
</tr>
</tbody>
</table>

Table 4: Covering of test set

<table>
<thead>
<tr>
<th>Corpus</th>
<th>align</th>
<th>Chinese</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Giza+</td>
<td>21.7%</td>
<td>36.0%</td>
<td></td>
</tr>
<tr>
<td>Pialign</td>
<td>23.6%</td>
<td>38.3%</td>
<td></td>
</tr>
<tr>
<td>2 Giza+</td>
<td>21.7%</td>
<td>36.1%</td>
<td></td>
</tr>
<tr>
<td>Pialign</td>
<td>23.8%</td>
<td>38.7%</td>
<td></td>
</tr>
<tr>
<td>3 Giza+</td>
<td>21.9%</td>
<td>36.6%</td>
<td></td>
</tr>
<tr>
<td>Pialign</td>
<td>23.9%</td>
<td>38.9%</td>
<td></td>
</tr>
</tbody>
</table>

In Table 3, we showed the total number of phrase pairs, the common phrase pairs of Giza++ and Pialign, the different phrase pairs of Giza++ and Pialign. In Table 4, we show the covering capacity of the phrase table. The covering capacity is defined as follows:

\[ c = \frac{\# \text{ of phrases both in test set and in phrase table}}{\# \text{ of phrases in test set}} \]

To note that, the test set was divided into unigram to 5-gram phrases.

From Table 3 we can find that the phrase table generated by Pialign is a little bigger than Giza++. Because we use -samps parameters to sample the bilingual parser tree repeatedly. In this experiment, we tuned this parameters from 1(default) to 80. At first, with the increment of the phrase table size, the performance grows at the same time. But after 20th sampling, the bias of sampling adds too many noise phrase pairs. Finally, we set this value to 20. With default value, Pialign only generated 389,982 phrase pairs (32.28% as the Giza++ did), but the performances are still comparable.

With the covering capacity, we can estimate the performance of the model. The result is the same with the translation result, which shows that Pialign is better than Giza++ in phrase extraction.

4.2. Results of translation
The result of translation outputs are shown in Table 5 and Table 6.

The result is confusing. After we tuned the parameters with HIT_dev, the result became worse. This may be caused by the mismatch between HIT_dev and HIT_train. The result also shows that although we continue to enlarge the size of
training data, the BLEU score may reduce on the contrary. These remind us that the model is also important.

**Table 5: Result without tuning**

<table>
<thead>
<tr>
<th>Corpus</th>
<th>align</th>
<th>BLEU%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Giza++</td>
<td>20.76</td>
</tr>
<tr>
<td></td>
<td>Pialign</td>
<td>20.80</td>
</tr>
<tr>
<td>2</td>
<td>Giza++</td>
<td>20.62</td>
</tr>
<tr>
<td></td>
<td>Pialign</td>
<td>21.20</td>
</tr>
<tr>
<td>3</td>
<td>Giza++</td>
<td>20.51</td>
</tr>
<tr>
<td></td>
<td>Pialign</td>
<td>20.54</td>
</tr>
</tbody>
</table>

**Table 6: Result with tuning**

<table>
<thead>
<tr>
<th>Corpus</th>
<th>align</th>
<th>BLEU%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Giza++</td>
<td>19.97</td>
</tr>
<tr>
<td></td>
<td>Pialign</td>
<td>19.70</td>
</tr>
<tr>
<td>2</td>
<td>Giza++</td>
<td>18.40</td>
</tr>
<tr>
<td></td>
<td>Pialign</td>
<td>19.66</td>
</tr>
<tr>
<td>3</td>
<td>Giza++</td>
<td>15.52</td>
</tr>
<tr>
<td></td>
<td>Pialign</td>
<td>15.10</td>
</tr>
</tbody>
</table>

### 4.3. Combination of two phrase table

We explored Giza++ and Pialign to extract phrases. In this section, we want to combine the two methods by merging two phrase tables using a linear interpolation method. For Giza++, the best result was achieved when we used Corpus1. For Pialign, the best result was achieved when we used Corpus2. So we combined the two phrase tables. The result without tuning is shown in Table 7. The parameter means the weight of Pialign.

**Table 7: Phrase Table Combination**

<table>
<thead>
<tr>
<th>parameter</th>
<th>BLEU%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>20.69</td>
</tr>
<tr>
<td>0.5</td>
<td>20.78</td>
</tr>
<tr>
<td>0.6</td>
<td>20.62</td>
</tr>
</tbody>
</table>

Compared with Table 7 and Table 5, we can draw a conclusion that phrase table combination can improve the performance of machine translation systems a little. Maybe due to the size of the training data, the result is not very clear to see the increment. And our combination method is only a linear interpolation method, which is naive for phrase table combination. We believe that a more complex strategy, such as some machine learning algorithms can improve the phrase table combination results.

### 4.4. Linguistic knowledge

In recently years, many researchers have focused on how to integrate linguistic knowledge into machine translation systems. In this work, part of speech was introduced to improve the machine translation systems. We used Stanford Log-linear Part-Of-Speech Tagger[11] to get the POS tag. Factored-based model of Moses was used to train a translation model. The result is shown in Table 8.

**Table 8: Linguistic features**

<table>
<thead>
<tr>
<th>system</th>
<th>With tuning</th>
<th>Without tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>19.97</td>
<td>20.76</td>
</tr>
<tr>
<td>+pos tag</td>
<td>18.53</td>
<td>16.63</td>
</tr>
</tbody>
</table>

As we can see that the result with POS tag is also not better than the baseline. We think that linguistic knowledge is a good research field to improve machine translation performance.

### 4.5. Official Results

We took part in the Olympics task(OLY)[12], and the final translations we submitted was generated by Pialign with corpus 2. And because of the bad performance of tuning, we submit out results without tuning. The final result was shown in Table 9.

**Table 9: Official results in BLEU**

<table>
<thead>
<tr>
<th>system</th>
<th>case+punc</th>
<th>no_case+no_punc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pialign-2</td>
<td>19.10</td>
<td>18.76</td>
</tr>
</tbody>
</table>

5. Conclusions and Future Work

In this paper, we explained our work in the IWSLT 2012 evaluation campaign. We compared two phrase extraction methods and tried to combine the two methods. The results show that the combination method can improve the result of MT systems.

In future, we will still try to study some other advanced combination methods to modify our system.

6. Acknowledgements

The work of this paper is funded by the project of National Natural Science Foundation of China (No. 61100093) and the project of National High Technology Research and Development Program of China (863 Program) (No. 2011AA01A207).

7. References


