The NTT Statistical Machine Translation System for IWSLT2005

Hajime Tsukada, Taro Watanabe, Jun Suzuki, Hitoto Kazawa, and Hideki Isozaki

NTT Communication Science Laboratories
{tsukada,taro,jun,kazawa,isozaki}@cslab.kecl.ntt.co.jp

Abstract

This paper reports the NTT statistical translation system participating in the evaluation campaign of IWSLT 2005. The NTT system is based on a phrase translation model and utilizes a large number of features with a log-linear model. We studied the various features recently developed in this research field and evaluate the system using supplied data as well as publicly available Chinese, Japanese, and English data. Despite domain mismatch, additional data helped improve translation accuracy.

1. Introduction

Recently, phrase-based translation combined with other features by log-linear models has become the standard technique for statistical machine translation. Shared task based workshops of machine translation including IWSLT and NIST Machine Translation Workshops showed which features effectively improve translation accuracy. However, it remains unclear whether using of these features all together with our system is helpful.

One unavoidable problem with statistical approaches is training data preparation. Since the amount of training data is generally limited, how to utilize similar monolingual or bilingual resources is an important research topic in statistical machine translation. In this evaluation campaign, we studied the use of a large number of reportedly effective features with our system and also evaluated both additional monolingual and bilingual corpus to improve translation accuracies.

2. Log-linear Models

Our system adopts the following log-linear decision rule to obtain the maximum likely translation:

\[
\hat{e}^t_1 = \arg \max_{e^t_1} \frac{1}{Z(f^t_1)} \exp \left( \sum_j \lambda_j f_j(e^t_1, f^t_1) \right),
\]

where \( f_j(e^t_1, f^t_1) \) represents a feature function and \( Z(f^t_1) \) denotes a normalization term. Feature function scaling factors \( \lambda_j \) are efficiently computed based either on the maximum likelihood criterion [1] or the minimum error rate criterion [2]. Our system adopts the latter criterion in the experiments.

One advantage of log-linear models is the ability to easily combine various features relating to translation models, language models, and lexical reordering models. The feature details are described in the following sections.

3. Features

3.1. Language Model Features

In statistical machine translation, improving language models strongly impacts translation accuracy. Especially recently, the power of long-span n-grams and the use of huge amounts of training data have been reported [3]. In this evaluation campaign, we combined several long n-gram language models as features.

To train language models from various corpora in different domains, corpus weighting is necessary to fit the trained language models to the test set domain. Log-linear models naturally provide weighting of a language model trained by each corpus.

We used the following long n-gram language models:
- 6-gram
- Class-based 9-gram
- Prefix-4 9-gram
- Suffix-4 9-gram

All n-grams are based on mixed casing. The prefix-4 (suffix-4) language model takes only 4-letter prefixes (suffixes) of English words. Prefix-4 (suffix-4) roughly means the word stem (inflectional endings). For example, “Would it be possible to ship it to Japan” becomes “Woul+ it be poss+ to ship it to Japa+” by prefix-4, and “Would it be +ible to ship it to Japan” by suffix-4, where “+” at the end or beginning of a word denotes deletion. Prefix-4 and suffix-4 are likely to contribute to word alignment and language modeling, respectively.

3.2. Phrase-based Features

Our system adopts a phrase-based translation model represented by phrase-based features, which are based on phrase
translation pairs extracted by the method proposed by Och and Ney [4].

First, many-to-many word alignment is set by using both one-to-many and many-to-one word alignments generated by GIZA++ toolkit. In the experiment, we used prefix-4 for word-to-word alignment. Using prefix-4 produced better translations than the original form in preliminary experiments.

Next, phrase pairs consistent with word alignment are extracted. The words in a legal phrase pair are only aligned to each other and not to words outside. Hereafter, we use \( \text{count}(\tilde{e}) \) and \( \text{count}(\tilde{f}, \tilde{e}) \) to denote the number of extracted phrase \( \tilde{e} \) and extracted phrase pair \((\tilde{f}, \tilde{e})\), respectively.

We used the following features based on extracted phrase pairs:

- Phrase translation probability \( \phi(\tilde{e} | \tilde{f}) \) and \( \phi(\tilde{f} | \tilde{e}) \), where
  \[
  \phi(\tilde{e} | \tilde{f}) = \frac{\text{count}(\tilde{f}, \tilde{e})}{\sum_j \text{count}(\tilde{f}, \tilde{e})}
  \]

- Frequency of phrase pairs \( \text{count}(\tilde{e}, \tilde{f}) \), \( \text{count}(\tilde{e}) \), and \( \text{count}(\tilde{f}) \)

- \( \chi^2 \) value and Dice coefficient of \( \tilde{f} \) and \( \tilde{e} \)

- Phrase extraction probability of source/target, i.e.,
  \[
  \frac{\# \text{ of extracted source/target phrases}}{\# \text{ of source/target phrases appearing in the corpus}}
  \]

- Phrase pair extraction probability, i.e.,
  \[
  \frac{\# \text{ of sentences phrase pairs extracted}}{\# \text{ of sentences phrase pairs appearing in the corpus}}
  \]

- Adjusted Dice coefficient, which is an extension of the measure proposed in [5], i.e.,
  \[
  \text{Dice}(\tilde{f}, \tilde{e}) \log(\text{count}(\tilde{f}, \tilde{e}) + 1)
  \]

3.3. Word-level Features

We used the following word-level features, where

\[
 w(f | e) = \frac{\text{count}(f, e)}{\sum_{f'} \text{count}(f', e)}.
\]

\( I \) is the number of words in the translation and \( J \) is the number of words in the input sentence.

- Lexical weight \( p_w(\tilde{f} | \tilde{e}) \) and \( p_w(\tilde{e} | \tilde{f}) \) [6], where
  \[
  p_w(\tilde{f} | \tilde{e}) = \max_a \prod_{j=1}^{J} \frac{1}{|\{i | (i, j) \in a\}|} \cdot \sum_{\forall(i, j) \in a} w(f_j | e_i)
  \]

- IBM Model 1 score \( p_{M1}(\tilde{f} | \tilde{e}) \) and \( p_{M1}(\tilde{e} | \tilde{f}) \), where
  \[
  p_{M1}(\tilde{f} | \tilde{e}) = \frac{1}{(I+1)^J} \prod_{j} \sum_{i} w(f_j | e_i)
  \]

- Viterbi IBM Model 1 score \( p_{M1'}(\tilde{f} | \tilde{e}) \) and \( p_{M1'}(\tilde{e} | \tilde{f}) \), where
  \[
  p_{M1'}(\tilde{f} | \tilde{e}) = \frac{1}{(I+1)^J} \prod_{j} \max_i w(f_j | e_i)
  \]

- Noisy OR gate \( p_{NOR}(\tilde{f} | \tilde{e}) \) and \( p_{NOR}(\tilde{e} | \tilde{f}) \) [7], where
  \[
  p_{NOR}(\tilde{f} | \tilde{e}) = \prod_j \left( 1 - \prod_i (1 - w(f_j | e_i)) \right)
  \]

- Deletion penalty \( p_{del}(\tilde{e}, \tilde{f}) \) where
  \[
  p_{del}(\tilde{e}, \tilde{f}) = \sum_j \text{del}(\tilde{e}_1, \tilde{f}_j)
  \]

\( \text{del}(\tilde{e}_1, \tilde{f}_j) = \begin{cases} 1 & i \text{ does not exist s.t. } w(\tilde{e}_i | \tilde{f}_j) > \text{threshold} \\ 0 & \text{otherwise.} \end{cases} \)

3.4. Lexical Reordering Features

We used the following features to control the reordering of phrases:

- Distortion model \( d(a_i - b_{i-1}) = \exp^{-|a_i - b_{i-1} - 1|} \), where \( a_i \) denotes the starting position of the foreign phrase translated into the \( i \)-th English phrase, and \( b_{i-1} \) denotes the end position of the foreign phrase translated into the \( (i - 1) \)-th English phrase [6].

- Right monotone model \( P_R(\tilde{e}, \tilde{f}) \) (and left monotone model \( P_L(\tilde{e}, \tilde{f}) \)) inspired by Och’s scheme [8], where
  \[
  P_R(\tilde{f}, \tilde{e}) = \frac{\text{count}_R}{\text{count}(\tilde{f}, \tilde{e})},
  \]
  and \( \text{count}_R \) denotes the number of right connected monotone phrases.

3.5. Other Features

The following additional features are used.

- number of words that constitute a translation
- number of phrases that constitute a translation
4. Decoder

The decoder is based on word graph [9] and uses a multi-pass strategy to generate n-best translations. It generates hypothesized translations in a left-to-right order by combining phrase translations for a source sentence. The first pass of our decoding algorithm generates a word graph, a compact representation of hypothesized translations, using a breadth-first beam search, as in [10][11][12][13]. Then, n-best translations are extracted from the generated word graph using $A^*$ search.

The search space for a beam search is constrained by restricting the reordering of source phrases. We have window size constraints that restrict the number of words skipped before selecting a segment of the source sequence [6][12]. An ITG-constraint [14] is also implemented that prohibits the extension of a hypothesis that violates ITG constraints, which will be useful for language pairs with drastic reordering, such as Japanese-to-English and Korean-to-English translations.

During the beam search stage, three kinds of pruning are performed to further reduce the search space [11]. First, observation pruning limits the number of phrase translation candidates to a maximum of $N$ candidates. Second, threshold pruning is performed by computing the most likely partial hypothesis and by discarding hypotheses whose probability is lower than the maximum score multiplied with a threshold. Third, histogram pruning is carried out by restricting the number of hypotheses to a maximum of $M$ candidates. Observation and threshold pruning are also applied to the back pointer to reduce the size of the word graph. In pruning hypotheses, future cost is also estimated on the fly and then integrated with the preceding score for beam pruning.

We estimated future cost as described in [13]. Exact costs for the phrase-based features and word level features can be calculated for each extracted phrase pair. For the language model features, their costs were approximated by considering the possible permutations of phrase pairs for a given input.

After generating a word graph, it is then pruned using the posterior probabilities of edges [15] to further reduce the number of duplicate translations for $A^*$ search. An edge is pruned if its posterior score is lower than the highest posterior score in the graph by a certain amount.

5. Experiments

To validate the use of the reportedly effective features, we conducted translation experiments using all features introduced in Section 3. Also, we conducted comparable experiments in both supplied and unrestricted data tracks to study the effectiveness of additional language resources.

<table>
<thead>
<tr>
<th>English data sets</th>
<th>Corpus size (words)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWSLT (supplied)</td>
<td>190,177</td>
</tr>
<tr>
<td>ATR</td>
<td>1,100,194</td>
</tr>
<tr>
<td>WEB</td>
<td>8,482,782</td>
</tr>
<tr>
<td>Gigaword</td>
<td>1,799,531,558</td>
</tr>
</tbody>
</table>

Table 1: Monolingual corpora for unrestricted data track

5.1. Corpus Preparation

To obtain comparable results for all source and target language pairs, we concentrated on tracks generating English, i.e., Japanese-to-English, Chinese-to-English, Arabic-to-English, and Korean-to-English.

The English parts of the corpora are tokenized using LDC’s standards. For Arabic, it is simply tokenized by splitting punctuation and then removing Arabic characters denoting “and”. For other languages, supplied segmentation is used. For unrestricted data tracks, Japanese is segmented using ChaSen 1, and Chinese is segmented using an LDC segmenter with lexicon entries gathered from supplied data and an LDC corpus. Test sets including ASR 1-best are also re-segmented in the same manner to maintain segmentation consistency.

We used mixed casing and prefix-4 form for word-to-word alignment in the phrase extraction. Also, mixed casing was used for training n-grams.

5.2. Language Models

6-gram language models and class-based/prefix-4/suffix-4 9-gram models trained by the SRI language modeling toolkit [16] were used in both supplied and unrestricted data tracks.

We used the following additional monolingual corpora for language models of unrestricted data tracks: (i) ATR Spoken Language Database publically available from ATR 2; (ii) Web pages crawled from discussion groups and FAQs about travel; and (iii) English Gigaword corpus from LDC.

As additional bilingual corpora for translation models of unrestricted data tracks, we used the ATR Spoken Language Database for Japanese-to-English translation and the two largest corpora in the LDC collection, LDC2004T08 and LDC2005T10, for Chinese-to-English translation. No additional resources were used for other language pairs. Tables 1 and 2 illustrate the data size of each corpus.

Using the monolingual corpora, a total of 10 n-grams were trained and used as a feature of log-linear models when decoding. Table 3 shows the output language perplexity of each n-gram used in the decoder. On the other hand, Table 4 shows the input language perplexity of the trigram trained by the supplied corpora. Tables 3 and 4 suggest that the ATR

1http://chasen.naist.jp
2http://www.red.atr.jp/product/index.html
Table 2: Bilingual corpora for unrestricted data track

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Language pairs</th>
<th>Corpus size (English words)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWSLT (supplied)</td>
<td>JE/CE/AE/KE</td>
<td>190,177</td>
</tr>
<tr>
<td>ATR</td>
<td>JE</td>
<td>1,334,852</td>
</tr>
<tr>
<td>LDC</td>
<td>CE</td>
<td>76,939,292</td>
</tr>
<tr>
<td>-</td>
<td>AE</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>KE</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4: Input language perplexity of trigram trained by supplied corpora

<table>
<thead>
<tr>
<th>Test sets</th>
<th>Japanese</th>
<th>Chinese</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IWSLT</td>
<td>ATR</td>
</tr>
<tr>
<td>devset1</td>
<td>16.9</td>
<td>29.5</td>
</tr>
<tr>
<td>devset2</td>
<td>17.6</td>
<td>32.9</td>
</tr>
<tr>
<td>testset</td>
<td>24.5</td>
<td>28.6</td>
</tr>
</tbody>
</table>

and IWSLT datasets are similar, WEB is closer to IWSLT than Gigaword, and that LDC is very different from IWSLT.

Since the collection is enormous in Gigaword, the vocabulary set is first limited to that observed in the English part of supplied corpus and the ATR database. Then for decoding, an actual n-gram language model is estimated on the fly by constraining the vocabulary set to that observed in a given test set.

5.3. Other Setups

Following one of the best systems [17] in IWSLT 2004, feature function scaling factors $\lambda_j$ are trained using NIST scores [18] in a loss function of minimum error rate training, and development set 1 (CSTAR) was used for it.

For Japanese and Korean, ITG constraints of lexical reordering were applied, and for Arabic and Chinese, simple window size constraints up to 7 were used.

5.4. Results

Table 5 summarizes the overall results of the supplied/unrestricted data tracks. The scores of the table are obtained by the comparable conditions for each language pair while some are not the same as those released by the organizer.

"unrestricted" denotes that monolingual corpora are unrestricted but bilingual corpora are restricted; "mbunrestricted" denotes that both monolingual and bilingual corpora are unrestricted.

The table shows that unrestricted data tracks consistently outperform restricted data tracks except for Japanese-to-English with ASR output. This may be because re-segmentation of the ASR output produces bad segmentation because of ASR errors.

"unrestricted" is inferior to "mbunrestricted" in Chinese-to-English translation whereas the former is better than the latter in Japanese-to-English translation. This may be because the additional bilingual resources are similar in Japanese-to-English but are different in Chinese-to-English as shown in Tables 3 and 4.

The overall results suggest that our feature design could not deal with domain mismatch of bilingual corpora but could deal with small mismatch of monolingual corpora. While Gigaword differs most from the supplied corpus in terms of perplexity, as shown in Table 3, its n-gram surprisingly contributes more to translation than other n-grams in terms of feature function scaling factors of log-linear models. It would be interesting to study this finding in more detail.

6. Conclusion

The NTT statistical machine translation system in the evaluation campaign is reported. A log-linear model naturally enabled weighting of various features including language models. As a result, we obtained competitive accuracies. The log-linear model effectively utilized n-grams trained by out-of-domain corpora, and we improved the translation accuracy of the supplied data.

These experiments simply used all available features. However, feature extraction may additionally improve translation accuracy. It is worth studying.

Compared to other sites, our results are better in terms of NIST scores but inferior in terms of BLEU scores. This is because feature function scaling factors are trained by a loss function based on NIST scores. We also doubt the overfitting of feature function scaling factors. We need to continue studying both training methods of the scaling factors and loss functions to improve other translation metrics as well as NIST scores.

7. Acknowledgements

Our training tool for phrase translation models is an extension of that provided by Philipp Koehn under a contract of MIT-NTT collaboration.

8. References


Table 3: Output language perplexity of n-grams for decoding

<table>
<thead>
<tr>
<th>Language pairs</th>
<th>Translation input</th>
<th>Training data</th>
<th>BLEU scores</th>
<th>NIST scores</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE</td>
<td>transcription</td>
<td>supplied m unrestricted</td>
<td>0.4350</td>
<td>9.1821</td>
</tr>
<tr>
<td></td>
<td></td>
<td>m unrestricted</td>
<td>0.4764</td>
<td>9.3674</td>
</tr>
<tr>
<td>CE</td>
<td>transcription</td>
<td>supplied m unrestricted</td>
<td>0.3275</td>
<td>8.0768</td>
</tr>
<tr>
<td></td>
<td></td>
<td>m unrestricted</td>
<td>0.4112</td>
<td>8.8418</td>
</tr>
<tr>
<td></td>
<td></td>
<td>m unrestricted</td>
<td>0.3943</td>
<td>8.6804</td>
</tr>
<tr>
<td></td>
<td>ASR 1-best</td>
<td>supplied m unrestricted</td>
<td>0.2739</td>
<td>6.5185</td>
</tr>
<tr>
<td></td>
<td></td>
<td>m unrestricted</td>
<td>0.2965</td>
<td>6.9416</td>
</tr>
<tr>
<td>JE</td>
<td>transcription</td>
<td>supplied m unrestricted</td>
<td>0.3669</td>
<td>7.9669</td>
</tr>
<tr>
<td></td>
<td></td>
<td>m unrestricted</td>
<td>0.3679</td>
<td>8.1207</td>
</tr>
<tr>
<td></td>
<td></td>
<td>m unrestricted</td>
<td>0.3932</td>
<td>8.6442</td>
</tr>
<tr>
<td></td>
<td>ASR 1-best</td>
<td>supplied m unrestricted</td>
<td>0.3881</td>
<td>8.3855</td>
</tr>
<tr>
<td></td>
<td></td>
<td>m unrestricted</td>
<td>0.3762</td>
<td>8.3502</td>
</tr>
<tr>
<td>KE</td>
<td>transcription</td>
<td>supplied m unrestricted</td>
<td>0.3218</td>
<td>7.8489</td>
</tr>
<tr>
<td></td>
<td></td>
<td>m unrestricted</td>
<td>0.3497</td>
<td>8.0160</td>
</tr>
</tbody>
</table>

Table 5: NTT results of evaluation campaign


