Large-scale Dictionary Construction via Pivot-based Statistical Machine Translation with Significance Pruning and Neural Network Features

Raj Dabre[1], **Chenhui Chu**[2], **Fabien Cromieres**[2], **Toshiaki Nakazawa**[2], **Sadao Kurohashi**[1]
[1]Graduate School of Informatics, Kyoto University
[2]Japan Science and Technology Agency
prajdabre@gmail.com, (chu,fabien,nakazawa)@pa.jst.jp, kuro@i.kyoto-u.ac.jp

Abstract

We present our ongoing work on large-scale Japanese-Chinese bilingual dictionary construction via pivot-based statistical machine translation. We utilize statistical significance pruning to control noisy translation pairs that are induced by pivoting. We construct a large dictionary which we manually verify to be of a high quality. We then use this dictionary and a parallel corpus to learn bilingual neural network language models to obtain features for reranking the n-best list, which leads to an absolute improvement of 5% in accuracy when compared to a setting that does not use significance pruning and reranking.

1 Introduction

Pivot-based statistical machine translation (SMT) (Wu and Wang, 2007) has been shown to be a possible way of constructing a dictionary for the language pairs that have scarce parallel data (Tsunakawa et al., 2009; Chu et al., 2015). The assumption of this method is that there is a pair of large-scale parallel data: one between the source language and an intermediate resource rich language (henceforth called pivot), and one between that pivot and the target language. We can use the source-pivot and pivot-target parallel data to develop a source-target term[1] translation model for dictionary construction.

Pivot-based SMT uses the log linear model as conventional phrase-based SMT (Koehn et al., 2007) does. This method can address the data sparseness problem of directly merging the source-pivot and pivot-target terms, because it can use the portion of terms to generate new terms. Small-scale experiments in (Tsunakawa et al., 2009) showed very low accuracy of pivot-based SMT for dictionary construction.[2]

This paper presents our study to construct a large-scale Japanese-Chinese (Ja-Zh) scientific dictionary, using large-scale Japanese-English (Ja-En) ($49.1M$ sentences and $1.4M$ terms) and English-Chinese (En-Zh) ($8.7M$ sentences and $4.5M$ terms) parallel data via pivot-based SMT. We generate a large pivot translation model using the Ja-En and En-Zh parallel data. Moreover, a small direct Ja-Zh translation model is generated using small-scale Ja-Zh parallel data. ($680k$ sentences and $561k$ terms). Both the direct and pivot translation models are used to translate the Ja terms in the Ja-En dictionaries to Zh and the Zh terms in the Zh-En dictionaries to Ja to construct a large-scale Ja-Zh dictionary (about $3.6M$ terms).

We address the noisy nature of pivoting large phrase tables by statistical significance pruning (Johnson et al., 2007). In addition, we exploit linguistic knowledge of common Chinese characters (Chu et al., 2013) shared in Ja-Zh to further improve the translation model. Large-scale experiments on scientific domain data indicate that our proposed method achieves high quality dictionaries which we manually verify to have a high quality.

Reranking the n-best list produced by the SMT decoder is known to help improve the translation quality given that good quality features are used (Och et al., 2004). In this paper, we use bilingual neural network language model features for reranking the n-best list produced by the pivot-based system which uses significance pruning, and achieve a 2.5% (absolute) accuracy improvement. Compared to a setting which uses neither significance pruning nor n-best list reranking the improvement in accu-

[1] In this paper, we call the entries in the dictionary terms. A term consists of one or multiple tokens.

[2] The highest accuracy evaluated based on the 1 best translation is 21.7% in (Tsunakawa et al., 2009).

29th Pacific Asia Conference on Language, Information and Computation pages 289 - 297
Shanghai, China, October 30 - November 1, 2015

racy is about 5% (absolute). We also use character based neural MT to eliminate the out-of-vocabulary (OOV) terms, which further improves the quality.

The rest of this paper is structured as follows: Section 2 reviews related work. Section 3 presents our dictionary construction using pivot-based SMT with significance pruning. Section 4 describe the bilingual neural language model features using a parallel corpus and the constructed dictionary for reranking the n-best list. Experiments and results are described in Section 5, and we conclude this paper in Section 6.

2 Related Work

Many studies have been conducted for pivot-based SMT. Utiyama and Isahara (2007) developed a method (sentence translation strategy) for cascading a source-pivot and a pivot-target system to translate from source to target using a pivot language. Since this results in multiplicative error propagation, Wu and Wang (2009) developed a method (triangulation) in which they combined the source-pivot and pivot-target phrase tables to obtain a source-target phrase table. They then combine the pivoted and direct tables (using source-target parallel corpora) by linear interpolation whose weights were manually specified. There is a method to automatically learn the interpolation weights (Sennrich, 2012) but it requires reference phrase pairs which are not easily available. Work on translation from Indonesian to English using Malay and Spanish to English using Portuguese (Nakov and Ng, 2009) as pivot languages worked well since the pivots had substantial similarity to the source languages. They used the multiple decoding paths (MDP) feature of the phrase-based SMT toolkit Moses (Koehn et al., 2007) to combine multiple tables which avoids interpolation. The issue of noise introduced by pivoting has not been seriously addressed and although statistical significance pruning (Johnson et al., 2007) has shown to be quite effective in a bilingual scenario, it has never been considered in a pivot language scenario.

(Tsunakawa et al., 2009) was the first work that constructs a dictionary for language pairs that are resource poor using pivot-based SMT, however the experiments were performed on small-scale data. Chu et al. (2015) conducted large-scale experiments and exploited the linguistic knowledge of common Chinese characters shared in Japanese-Chinese (Chu et al., 2013) to improve the translation model.

N-best list reranking (Och et al., 2004; Sutskever et al., 2014) is known to improve the translation quality if good quality features are used. Recently, (Cho et al., 2014) and (Bahdanau et al., 2014) have shown that recurrent neural networks can be used for phrase-based SMT whose quality rivals the state of the art. Since the neural translation models can also be viewed as bilingual language models, we use them to obtain features for reranking the n-best lists produced by the pivot-based system.

3 Dictionary Construction via Pivot-based SMT

Figure 1 gives an overview of our construction method. Phrase-based SMT (Koehn et al., 2007) is the basis of our method. We first generate Ja-Zh (source-target), Ja-En (source-pivot) and En-Zh (pivot-target) phrase tables from parallel data respectively. The generated Ja-Zh phrase table is used as the direct table. Using the Ja-En and En-Zh phrase tables, we construct a Ja-Zh pivot phrase table via En. The direct and pivot tables are then combined and used for phrase-based SMT to the Ja terms in the Ja-En dictionaries to Zh and the Zh terms in the Zh-En dictionaries to Ja to construct a large-scale Ja-Zh dictionary. In addition, we use common Chinese characters to generate Chinese character features for the phrase tables to improve the SMT performance.

3.1 Pivot Phrase Table Generation

We follow the phrase table triangulation method (Wu and Wang, 2007) to generate the pivot phrase table. This method generates a source-target phrase table via all their shared pivot phrases in the source-pivot and pivot-target tables. The formulae for generating the inverse phrase translation probabilities and direct lexical weightings, $\phi(f|e)$ and $lex(f|e)$ are given below. Inverting the positions of **e** and **f** give the formulae for the direct probabilities and weightings, $\phi(e|f)$ and $lex(e|f)$.

$$\phi(f|e) = \sum_{p_i} \phi(f|p_i) * \phi(p_i|e) \qquad (1)$$

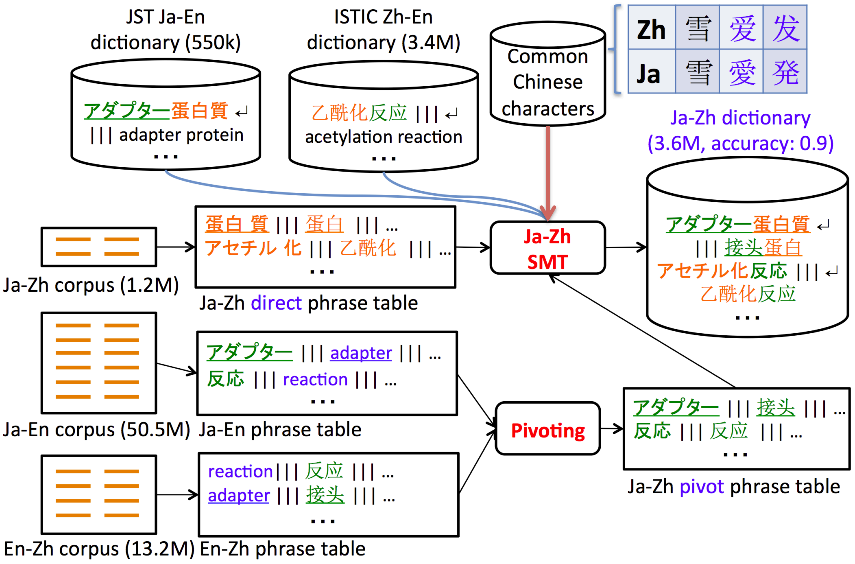

JST Ja-En dictionary (550k)
アダプター蛋白質 ↵ ||| adapter protein
...
ISTIC Zh-En dictionary (3.4M)
乙酰化反应 ||| ↵ acetylation reaction
...
Common Chinese characters
Zh 雪 爱 发
Ja 雪 愛 発
Ja-Zh dictionary (3.6M, accuracy: 0.9)
アダプター蛋白質 ↵ ||| 接头蛋白
アセチル化反応 ||| ↵ 乙酰化反应
...
Ja-Zh corpus (1.2M)
蛋白 質 ||| 蛋白 ||| ...
アセチル 化 ||| 乙酰化 ||| ...
...
Ja-Zh direct phrase table
Ja-Zh SMT
Ja-En corpus (50.5M)
アダプター ||| adapter ||| ...
反応 ||| reaction ||| ...
...
Ja-En phrase table
Pivoting
アダプター ||| 接头 ||| ...
反応 ||| 反应 ||| ...
...
Ja-Zh pivot phrase table
En-Zh corpus (13.2M)
reaction||| 反应 ||| ...
adapter ||| 接头 ||| ...
...
En-Zh phrase table

Figure 1: Overview of our dictionary construction method.

$$lex(f|e,a) = \sum_{p_i} lex(f|p_i,a_1) * lex(p_i|e,a_2) \quad (2)$$

where a_1 is the alignment between phrases f (source) and p_i (pivot), a_2 is the alignment between p_i and e (target) and a is the alignment between e and f. Note that the lexical weightings are calculated in the same way as the phrase probabilities. Our results might be further improved if we used more sophisticated approaches like the cross-language similarity method or the method which uses pivot induced alignments (Wu and Wang, 2007).

As pivoting induces a very large number of phrase pairs, we prune all pairs with inverse phrase translation probability less than 0.001. This manually specified threshold is simple, and works in practice but is not statistically motivated.

3.2 Combination of the Direct and Pivot Phrase Tables

To combine the direct and pivot phrase tables, we make use of the MDP method of the phrase-based SMT toolkit Moses (Koehn et al., 2007), which has been shown to be an effective method (Nakov and Ng, 2009). MDP, which uses all the tables simultaneously while decoding, ensures that each pivot table is kept separate and translation options are collected from all the tables.

3.3 Exploiting Statistical Significance Pruning for Pivoting

Consider a source-pivot phrase pair (X,Y) and a pivot-target phrase pair (Y,Z). If Y is a bad translation of X and Z is a bad translation of Y, then the induced pair (X,Z) will also be a bad pair. The phrase pair extraction processes in phrase-based SMT often result in noisy phrase tables, which when pivoted give even noisier tables. Statistical significance pruning (Johnson et al., 2007) is known to eliminate a large amount of noise and thus we used it to prune our tables before pivoting. We used the $\alpha + \epsilon$ threshold which is based on the parallel corpus size and shown to be optimal.

Although the optimal thresholds for a pivot based MT setting might be different, currently we consider only the $\alpha + \epsilon$ threshold which is determined to be the best by (Johnson et al., 2007). Exhaustive testing using various thresholds will be performed and reported in the future. The negative log probability of the p-value (also called significance value) of the phrase pair is computed and the pair is retained if this exceeds the threshold. It is possible that all phrase pairs for a source phrase might be pruned leading to an out-of-vocabulary (OOV) problem. To remedy this we retain the top 5 phrase pairs (according to inverse translation probability) for such a phrase. We tried 3 different settings: Prune source-pivot table only (labeled “Pr:S-P”), Prune pivot-target table only (labeled “Pr:P-T”) and Prune both tables (labeled “Pr:Both”). We discuss the effects of each setting in Section 5.2.4.

3.4 Chinese Character Features

Ja-Zh shares Chinese characters. Because many common Chinese characters exist in Ja-Zh, they have been shown to be very effective in many Ja-Zh natural language processing (NLP) tasks (Chu et al., 2013). In this paper, we compute Chinese character features for the phrase pairs in the translation models, and integrate these features in the log-linear model for decoding. In detail, we compute following two features for each phrase pair:

$$CC\_ratio = \frac{Ja\_CC\_num + Zh\_CC\_num}{Ja\_char\_num + Zh\_char\_num} \tag{3}$$

$$CCC\_ratio = \frac{Ja\_CCC\_num + Zh\_CCC\_num}{Ja\_CC\_num + Zh\_CC\_num} \tag{4}$$

where $char\_num$, CC_num and CCC_num denote the number of characters, Chinese characters and common Chinese characters in a phrase respectively. The common Chinese character ratio is calculated based on the Chinese character mapping table in (Chu et al., 2013). We simply add these two scores as features to the phrase tables and use these tables for tuning and testing.

A combination of pivoting, statistical significance pruning and Chinese character features is used to construct the high quality large scale dictionary. One can use this dictionary as an additional component in an MT system. In our case we use it to generate features for N-best list reranking (next section).

4 N-best List Reranking using Neural Features

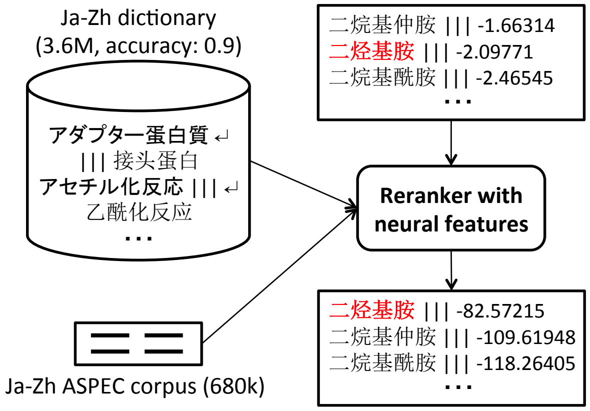


Figure 2: Using neural features for reranking.

The motivation behind n-best list reranking is simple: It is quite common for a good translation candidate to be ranked lower than a bad translation candidate. However, it might be possible to use additional features to rerank the list of candidates in order to push the good translation to the top of the list. Figure 2 gives a simple description of the n-best list reranking procedure using neural features. Using the Ja-Zh dictionary constructed using the methods specified in Section 3 and the Ja-Zh ASPEC corpus we train 4 neural translation models. For each translation direction we train a character based model using the dictionary and corpus separately (2 directions and 2 corpora lead to 4 models). It is important to note that although the dictionary is automatically created and is noisy, neural networks are quite robust and can regulate the noise quite effectively. This claim will be validated by our results (see Section 5.2.4). We use the freely available toolkit for neural MT, GroundHog[3], which contains an implementation of the work by (Bahdanau et al., 2014). After training a neural translation model it can be used either to translate an input sentence or it can be used to produce a score given an input sentence and a candidate translation. In the latter case, the neural translation model can be viewed as a **bilingual language model**.

One major limitation of neural network based models is that they are very slow to train in case of large vocabularies. It is possible to learn character based models but such models are not suited for extremely long sequences. In the case of Japanese and Chinese, however, since both languages use Chinese characters the character sequences are not too long and thus it makes sense to use character based MT here. Since the number of characters is quite smaller compared to the number of words, the training is quite fast. Ultimately, character based MT is always worse than word based MT and so, in this work we only use the character based neural MT models to obtain features for n-best list reranking. We also use

[3] https://github.com/lisa-groundhog/GroundHog

these models to perform character based translation of untranslated words and avoid OOVs.
The procedure we followed to perform reranking is given below. A decoder always gives n-best lists when performing tuning and testing. To learn reranking weights, we use the n-best list, for the tuning/development set, corresponding to the run with the highest evaluation metric score (BLEU in our case).

1. For each input term in the tuning set:
   (a) Obtain 4 neural translation scores for each translation candidate.
   (b) Append the 4 scores to the list of features for the candidate.
2. Use **kbmira**[4] to learn feature weights using the modified n-best list and the references for the tuning set.
3. Charater level BLEU as well as word level BLEU are used as reranking metric.
4. For each input term in the test set:
   (a) Obtain 4 neural translation scores for each translation candidate and append them to the list of features for that candidate.
   (b) Perform the linear combination of the learned weights and the features to get a model score.
5. Sort the n-best list for the test set using the calculated model scores (highest score is the best translation) to obtain the reranked list.

We also try another reranking method by treating it as a classification task using the support vector machine (SVM) toolkit.[5] When evaluating dictionaries, the translation is either correct or incorrect which is unlike sentence translation evaluation. We thus learn a SVM using the development set n-best list and the references to learn a classifier which is able to differentiate between a correct and an incorrect translation. The method we used for reranking is:

1. For each input term in the tuning set:
   (a) Obtain 4 neural translation scores for each translation candidate.
   (b) Append the 4 scores to the list of features for the candidate.
   (c) Generate classification label for candidate by comparing it with the reference.
2. Learn SVM classifier using the constructed training set.
3. For each input term in the test set:
   (a) Obtain 4 neural translation scores for each translation candidate and append them to the list of features for that candidate.
   (b) Use the SVM model to perform classification but give the probability scores instead of labels.
4. Sort the n-best list for the test set using the calculated probability scores (highest score is the best translation) to obtain the reranked list.

If there are any OOVs in the reranked n-best list then we replace them with the translation obtained using the above mentioned character based neural models (in the Ja-Zh direction).

5 Experiments

We describe the data sets, experimental settings and evaluations of the results below.

5.1 Training data

We used following two types of training data:

- Bilingual dictionaries: we used general domain Ja-En, En-Zh and Ja-Zh dictionaries (i.e. Wikipedia title pairs and EDR[6]), and the scientific dictionaries provided by the Japan Science and Technology Agency (JST)[7] and the Institute of Science and Technology information of China (ISTIC)[8] (called the JST dictionary and ISTIC dictionary hereafter), containing $1.4M$, $4.5M$ and $561k$ term pairs respectively. Table 1

[4] We used the K-best batch MIRA in the Moses decoder to learn feature weights.

[5] https://www.csie.ntu.edu.tw/cjlin/libsvm/

[6] https://www2.nict.go.jp/out-promotion/techtransfer/EDR/J_index.html

[7] http://www.jst.go.jp

[8] http://www.istic.ac.cn

Language	Name	Domain	Size
Ja-En	wiki_title	general	361,016
	med_dic	medicine	54,740
	EDR	general	491,008
	JST_dic	science	550,769
En-Zh	wiki_title	general	151,338
	med_dic	medicine	48,250
	EDR	general	909,197
	ISTIC_dic	science	3,390,792
Ja-Zh	wiki_title	general	175,785
	med_dic	medicine	54,740
	EDR	general	330,796

Table 1: Statistics of the bilingual dictionaries used for training.

Language	Name	Size
Ja-En	LCAS	3,588,800
	abst_title	22,610,643
	abst_JICST	19,905,978
	ASPEC	3,013,886
En-Zh	LCAS	6,090,535
	LCAS_title	1,070,719
	ISTIC_pc	1,562,119
Ja-Zh	ASPEC	680,193

Table 2: Statistics of the parallel corpora used for training (All the corpora belong to the general scientific domain, except for ISTIC_pc that is a computer domain corpus).

shows the statistics of the bilingual dictionaries used for training.

- Parallel corpora: the scientific Ja-En, En-Zh and Ja-Zh corpora we used were also provided by JST and ISTIC, containing $49.1M$, $8.7M$ and $680k$ sentence pairs respectively. Table 2 shows the statistics of parallel corpora used for training. Among which ISTIC_pc was provided by ISTIC, and the others were provided by JST.

5.2 Evaluation

5.2.1 Tuning and Testing data

We used the terms with two reference translations[9] in the Ja-Zh Iwanami biology dictionary (5,890 pairs) and the Ja-Zh life science dictionary (4,075 pairs) provided by JST. Half of the data in each dictionary was used for tuning (4,983 pairs), and the other half for testing (4,982 pairs). The evaluation scores on the test set give an idea of the quality of the constructed dictionary.

5.2.2 Settings

In our experiments, we segmented the Chinese and Japanese data using a tool proposed by Shen et al. (2014) and JUMAN (Kurohashi et al., 1994) respectively. For decoding, we used Moses (Koehn et al., 2007) with the default options. We trained a word 5-gram language model on the Zh side of all the En-Zh and Ja-Zh training data ($14.4M$ sentences) using the SRILM toolkit[10] with interpolated Keneser-Ney discounting. Tuning was performed by minimum error rate training which also provides us with the n-best lists used to learn reranking weights.

As a baseline, we compared following three methods for training the translation model:

- Direct: Only use the Ja-Zh data to train a direct Ja-Zh model.
- Pivot: Use the Ja-En and En-Zh data for training Ja-En and En-Zh models, and construct a pivot Ja-Zh model using the phrase table triangulation method.
- Direct+Pivot: Combine the direct and pivot Ja-Zh models using MDP.

We further conducted experiments using different significance pruning methods described in Section 3.3 and compared the following:

- Direct+Pivot (Pr:S-P): Pivoting after pruning the source-pivot table.
- Direct+Pivot (Pr:P-T): Pivoting after pruning the pivot-target table.
- Direct+Pivot (Pr:Both): Pivoting after pruning both the source-pivot and pivot-target tables.

We also conducted additional experiments using the Chinese character features (labeled +CC) (described in 3.4), but we only report the scores on Direct+Pivot (Pr:P-T), which is the best setting (thus labeled BS) for constructing the dictionary. Finally, using the

[9] Different terms are annotated with different number of reference translations in these two dictionaries.

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

BS, we translated the Ja terms in the JST ($550k$) dictionary to Zh and the Zh terms in the ISTIC ($3.4M$) dictionary to Ja, and constructed the Ja-Zh dictionary. The size of the constructed dictionary is $3.6M$ after discarding the overlapped term pairs in the two translated dictionaries. We then used this dictionary along with the Ja-Zh ASPEC parellel corpus to rerank the n-best list of the BS using the methods mentioned in Section 4. The following scores are reported:

- BS+RRCBLEU: Using character BLEU to rerank the n-best list.
- BS+RRWBLEU: Using word BLEU to rerank the n-best list.
- BS+RRSVM: Using SVM to rerank the n-best list.

This is followed by substituting the OOVs with the character level translations using the learned neural translation models (which we label as +OOVsub).

5.2.3 Evaluation Criteria

Following (Tsunakawa et al., 2009), we evaluated the accuracy on the test set using three metrics: 1 best, 20 best and Mean Reciprocal Rank (MRR)(Voorhees, 1999). In addition, we report the BLEU-4 (Papineni et al., 2002) scores that were computed on the word level.

5.2.4 Results of Automatic Evaluation

Table 3 shows the evaluation results. We also show the percentage of OOV terms,[11] and the accuracy with and without OOV terms respectively. In general, we can see that Pivot performs better than Direct, because the data of Ja-En and En-Zh is larger than that of Ja-Zh. Direct+Pivot shows better performance than either method.

Different pruning methods show different performances, where Pr:P-T improves the accuracy, while the other two not. To understand the reason for this, we also investigated the statistics of the pivot tables produced by different methods. Table 4 shows the statistics. We can see that compared to the other two pruning methods, Pr:P-T keeps the number of source phrases, which leads a lower OOV rate. It also prunes the number of average translations for each source phrase to a more reasonable number, which allows the decoder to make better decisions. Although the average number of translations for the Pr:Both setting is the smallest, it shows worse performance compared to Pr:P-T method. We suspect the reason for this is that many pivot phrases are pruned by Pr:Both, leading to fewer phrase pairs induced by pivoting. Augmenting with +CC leads to further improvements, and substituting the OOVs using their character level translation gives slightly better performance.

Method	Size	# src phrase	# avg trans
w/o pruning	29G	24,228	10,451
Pr:S-P	16G	19,502	7,058
Pr:P-T	5.5G	24,226	1,744
Pr:Both	2.8G	19,502	1,069

Table 4: Statistics of the pivot phrase tables (for tuning and test sets combined).

The most noteworthy results are obtained when reranking is performed using the bilingual neural language model features. BS+RRCBLEU, which uses character BLEU as a metric, performs almost as well as BS+RRWBLEU which uses word BLEU. There might be a difference in the BLEU scores of these 2 settings but the crucial aspect of dictionary evaluation is the accuracy regarding which there is no notable difference between them. We expected that since reranking using SVM, which focuses on accuracy and not BLEU, would yield better results but it might be the case that the training data obtained from the n-best lists is not very reliable. Finally, substuting the OOVs from the reranked lists further boosts the accuracies and although the increment is slight the OOV rate goes down to 0%. It is important to understand that the 20 best accuracy is 73% in the best case which means that if reranking is proper then it is possible to boost the accuracies by approximately 15%.

5.2.5 Results of Manual Evaluation

We manually investigated the terms, whose top 1 translation was evaluated as incorrect according to our automatic evaluation method. Based on our investigation, nearly 75% of them were actually correct translations. They were undervalued because

[11] An OOV term contains at least one OOV word.

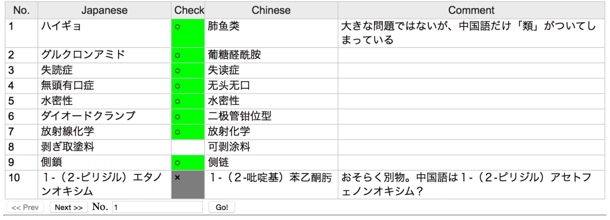
Method	BLEU-4	OOV term	Accuracy w/ OOV 1 best	 20 best	 MRR	Accuracy w/o OOV 1 best	 20 best	 MRR
Direct	40.64	26%	0.3697	0.5255	0.4258	0.4978	0.7082	0.5736
Pivot	52.32	8%	0.4938	0.7258	0.5730	0.5361	0.7880	0.6220
Direct+Pivot	53.69	8%	0.5088	0.7360	0.5902	0.5522	0.7987	0.6405
Direct+Pivot (Pr:S-P)	52.30	12%	0.4944	0.6881	0.5649	0.5589	0.7779	0.6386
Direct+Pivot (Pr:P-T)	55.44	8%	0.5267	0.7278	0.5990	0.5716	0.7898	0.6500
Direct+Pivot (Pr:Both)	49.71	12%	0.4591	0.6766	0.5391	0.5189	0.7649	0.6094
Direct+Pivot (Pr:P-T)+CC = [BS]	55.86	8%	0.5303	0.7260	0.6005	0.5755	0.7878	0.6517
BS+OOVsub	55.38	0%	0.5325	0.7300	0.6033	0.5325	0.7300	0.6033
BS+RRCBLEU	57.78	8%	0.5568	0.7260	0.6222	**0.6042**	**0.7878**	**0.6752**
BS+RRWBLEU	**58.55**	8%	0.5566	0.7260	0.6218	0.6040	**0.7878**	0.6748
BS+RRSVM	55.28	8%	0.5472	0.7260	0.6147	0.5938	**0.7878**	0.6670
BS+RRCBLEU+OOVsub	57.25	0%	**0.5590**	**0.7300**	**0.6249**	0.5590	0.7300	0.6249
BS+RRWBLEU+OOVsub	58.00	0%	0.5588	**0.7300**	0.6246	0.5588	0.7300	0.6246
BS+RRSVM+OOVsub	54.85	0%	0.5494	**0.7300**	0.6174	0.5494	0.7300	0.6174

Table 3: Evaluation results.

they were not covered by the reference translations in our test set. Taking this observation into consideration, the actual 1 best accuracy is about 90%. Automatic evaluation tends to greatly underestimate the results because of the incompleteness of the test set.

5.3 Evaluating the Large Scale Dictionary

As mentioned before the setting Direct+Pivot (Pr:P-T)+CC was used to translate the Ja terms in the JST ($550k$) dictionary to Zh and the Zh terms in the ISTIC ($3.4M$) dictionary to Ja so as to construct the Ja-Zh dictionary. The size of the constructed dictionary is $3.6M$ after discarding the overlapped term pairs in the two translated dictionaries. Since we had no references to automatically evaluate this massive dictionary, we evaluated its accuracy by humans. We asked 4 Ja-Zh bilingual speakers to evaluate 100 term pairs, which were randomly selected the constructed dictionary. Figure 3 shows the web interface used for human evaluation. It allows the evaluators to correct errors and well as leave subjective comments, which can be used to refine our methods. The evaluation results indicate that the 1 best accuracy is about 90%, which is consistent with the manual evaluation results on the test set.



No.	Japanese	Check	Chinese	Comment
1	ハイギョ	○	肺鱼类	大きな問題ではないが、中国語だけ「類」がついてしまっている
2	グルクロンアミド	○	葡糖醛酰胺	
3	失読症	○	失读症	
4	無頭有口症	○	无头无口	
5	水密性	○	水密性	
6	ダイオードクランプ	○	二极管钳位型	
7	放射線化学	○	放射化学	
8	剥ぎ取塗料		可剥涂料	
9	側鎖	○	侧链	
10	1-（2-ピリジル）エタノンオキシム	×	1-（2-吡啶基）苯乙酮肟	おそらく別物。中国語は1-（2-ピリジル）アセトフェノンオキシム？

Figure 3: Human evaluation web interface.

6 Conclusion and Future Work

In this paper, we presented a dictionary construction method via pivot-based SMT with significance pruning, chinese character knowledge and bilingual neural network language model based features reranking. Large-scale Ja-Zh experiments show that our method is quite effective. Manual evaluations showed that 90% of the terms are correctly translated, which indicates a high practical utility value of the dictionary. We plan to make the constructed dictionary available to the public in near future, and hope that crowdsourcing could be further used to improve it.

We observed that the weights learned for the neural features and found out that the highest weight was assigned to the feature obtained using the model learned using this dictionary. And since reranking did improve the accuracies on the test set, it is quite evident that this dictionary is of a fairly high quality. In the future we plan to try an iterative process, where we rerank the n-best list of this massive dictionary to get an improved dictionary on which we learn a better neural bilingual language model for reranking.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Chenhui Chu, Toshiaki Nakazawa, Daisuke Kawahara, and Sadao Kurohashi. 2013. Chinese-japanese machine translation exploiting chinese characters. *ACM Transactions on Asian Language Information Processing (TALIP)*, 12(4):16:1–16:25.

Chenhui Chu, Raj Dabre, Toshiaki Nakazawa, and Sadao Kurohashi. 2015. Large-scale japanese-chinese scientific dictionary construction via pivot-based statistical machine translation. In *Proceedings of the 21st Annual Meeting of the Association for Natural Language Processing (NLP 2015)*, pages 99–102, Kyoto, Japan, Match.

Howard Johnson, Joel Martin, George Foster, and Roland Kuhn. 2007. Improving translation quality by discarding most of the phrasetable. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 967–975, Prague, Czech Republic, June. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL*, pages 177–180.

Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. 1994. Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of the International Workshop on Sharable Natural Language*, pages 22–28.

Preslav Nakov and Hwee Tou Ng. 2009. Improved statistical machine translation for resource-poor languages using related resource-rich languages. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1358–1367, Stroudsburg, PA, USA. Association for Computational Linguistics.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 161–168, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318.

Rico Sennrich. 2012. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 539–549, Stroudsburg, PA, USA. Association for Computational Linguistics.

Mo Shen, Hongxiao Liu, Daisuke Kawahara, and Sadao Kurohashi. 2014. Chinese morphological analysis with character-level pos tagging. In *Proceedings of ACL*, pages 253–258.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215.

Takashi Tsunakawa, Naoaki Okazaki, Xiao Liu, and Jun'ichi Tsujii. 2009. A chinese-japanese lexical machine translation through a pivot language. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(2):9:1–9:21, May.

Masao Utiyama and Hitoshi Isahara. 2007. A comparison of pivot methods for phrase-based statistical machine translation. In *in Proceedings of the conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (NAACL-HLT*, pages 484–491.

Ellen M. Voorhees. 1999. The TREC-8 question answering track report. In *Proceedings of the Eighth TExt Retrieval Conference (TREC-8)*, pages 77–82.

Hua Wu and Haifeng Wang. 2007. Pivot language approach for phrase-based statistical machine translation. *Machine Translation*, 21(3):165–181, September.

Hua Wu and Haifeng Wang. 2009. Revisiting pivot language approach for machine translation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL '09, pages 154–162, Stroudsburg, PA, USA. Association for Computational Linguistics.