Performance of Japanese-to-Indonesian Machine Translation on Different Models

Cosmas Krisna Adiputra†, Yuki Arase‡,
†Division of Electronic and Information Engineering, School of Engineering, Osaka University
‡Graduate School of Information Science and Technology, Osaka University

{cosmas.a, arase}@ist.osaka-u.ac.jp

1 Introduction

As the relationship between Japan and Indonesia gets stronger in various fields such as education, economy, culture, research, and trade, the need of automatic machine translation is increasing. However, there is still little research effort on Japanese-Indonesian machine translation. Also, there is no standard Japanese-Indonesian parallel corpora available to build a decent translation model. In this study, we collect as many Japanese-Indonesian parallel data as possible and train state-of-the-art machine translation models in order to investigate their performance on this specific language pair.

Indonesian or *Bahasa Indonesia* is the official language of Indonesia. Indonesian is an SVO (Subject-Verb-Object) language that uses the same alphabet, syntax, and punctuations with English. In addition, Indonesian affixes are important because slightly different affixes may have very different meanings.

On the other hand, Japanese is an SOV language which uses kanji (chinese characters), hiragana, and katakana in its writing system. Japanese has also an extensive grammatical system to express politeness and formality.

Considering these characteristics of Indonesian and Japanese, machine translation between them has lots of challenges due to their linguistic differences. Simon and Purwarianti [5] build a Japanese-Indonesian statistical machine translation (SMT) system, however, their data was limited as only 500 parallel sentence pairs[*1]. Purwarianti et al. [9] also build a Indonesian-to-Japanese SMT system, but they use English as a pivot and no direct translation is considered.

We examine the effectiveness of the state-of-the-art machine translation models in order to understand problems and challenges on this language pair. Specifically, we evaluate SMT and neural machine translation (NMT) models. NMT applies neural network in translation and has begun to show promising results for several resourceful languages like English-French and English-German. NMT requires a large scale parallel corpora. Therefore, we collect a considerable amount of parallel corpora that are publicly available and use them to train the SMT and NMT models. Experiment results show that SMT achieves the best BLEU score. We find that vocabulary size due to rich affixes in Indonesian is a challenge for NMT.

[*1] confirmed by the authors

2 Machine Translation Models

In this section, we briefly review translation systems that we compare in our experiments.

2.1 Statistical Machine Translation

The goal of an SMT system is to find translation $\boldsymbol{f}$ given source sentence $\boldsymbol{e}$ by maximizing:

$$p(\boldsymbol{f}|\boldsymbol{e}) \propto p(\boldsymbol{e}|\boldsymbol{f})p(\boldsymbol{f}), \tag{1}$$

where $p(\boldsymbol{e}|\boldsymbol{f})$ and $p(\boldsymbol{f})$ are called *translation model* and *language model*, respectively [7]. Generally, the translation model takes the log-linear form in recent SMT systems with features and corresponding weights:

$$\log p(\boldsymbol{f}|\boldsymbol{e}) = \sum_{i=1}^{N} w_i f_i(\boldsymbol{f}, \boldsymbol{e}) + \log Z(\boldsymbol{e}), \tag{2}$$

where f_i and w_i are the i-th feature and weight, respectively. The normalization constant, $Z(e)$, does not depend on the weights. The weights are optimized to maximize the BLEU score on a development set.

2.2 Neural Machine Translation

2.2.1 Encoder-Decoder Model

Recent NMT models use the encoder-decoder framework proposed by [2], where encoder and decoder are realized using Recurrent Neural Network

(RNN). The encoder reads a source sentence and encodes it into a fixed-length vector, called context vector. The decoder generates translation by decoding the context vector into a sentence of variable length.

RNNs use recursion to compress a sequence of input symbols into a context vector. Assume at step t that we have a vector $\boldsymbol{h_{t-1}}$ which is the history of all previous symbols. The RNN will compute its internal state, $\boldsymbol{h_t}$ which compresses all the previous symbols ($\boldsymbol{x_1}$,$\boldsymbol{x_2}$,...,$\boldsymbol{x_{t-1}}$) including the current symbol $\boldsymbol{x_t}$ by calculating

$$\boldsymbol{h_t} = \phi_\theta(\boldsymbol{x_t}, \boldsymbol{h_{t-1}}), \tag{3}$$

where ϕ_θ is an activation function parametrized by $\boldsymbol{\theta}$ which takes the current symbol $\boldsymbol{x_t}$ and the history $\boldsymbol{h_{t-1}}$ as its input. After reading the end of the sequence, the hidden state of the RNN, $\boldsymbol{h_{len(x)}}$ is the context vector $\boldsymbol{c}$ of the whole input sequence.

The decoder is another RNN which is trained to generate the output sequence by predicting the next symbol $\boldsymbol{y_t}$ given the hidden state $\boldsymbol{h_t}$. Unlike the encoder, both $\boldsymbol{y_t}$ and $\boldsymbol{h_t}$ are also influenced by the previous predicted symbol $\boldsymbol{y_{t-1}}$ and the context vector $\boldsymbol{c}$ of the input sequence. Therefore, the hidden state of the decoder at step t is calculated by,

$$\boldsymbol{h_t} = \phi_{\theta'}(\boldsymbol{y_{t-1}}, \boldsymbol{h_{t-1}}, \boldsymbol{c}), \tag{4}$$

and the conditional distribution of the next symbol, $\boldsymbol{y_t}$ is computed by

$$P(\boldsymbol{y_t}|\boldsymbol{y_{t-1}}, \boldsymbol{y_{t-2}}, ..., \boldsymbol{y_1}, \boldsymbol{c}) = \psi(\boldsymbol{y_{t-1}}, \boldsymbol{h_{t-1}}, \boldsymbol{c}), \tag{5}$$

where ψ produces valid probabilities, e.g. with a softmax. The encoder and decoder are jointly trained to maximize the conditional log-likelihood. A parallel corpus Ω with N pairs and each sample in the corpus (X^n, Y^n) of source and target sentences are required. Given any pair from the corpus, the NMT model computes the conditional log-probability of Y^n given X^n: $\log P(Y^n|X^n, \boldsymbol{\theta})$ and log-likelihood of the whole training corpus as

$$\mathcal{L}(\Omega, \theta) = \frac{1}{N}\sum_{n=1}^{N} \log P(Y^n|X^n, \boldsymbol{\theta}). \tag{6}$$

The parameters $\boldsymbol{\theta}$ are tuned by maximizing Eq. 6.

2.2.2 Attention Mechanism

As variations of RNN that overcomes its limitations, long short term memory [4] and gated recurrent unit (GRU) [3] are commonly used in recent NMT systems. In addition, Bahdanau et al. [1] show that encoding input sequence in bi-direction improves translation quality, which is called as bidirectional RNN.

Attention mechanism has been shown to significantly improve translation quality in NMT, which was first introduced in [1]. Attention mechanism learns soft alignment between the source and target. It variates the value of context vector $\boldsymbol{c}$ by giving weight to hidden states of the encoder. An output sequence $\boldsymbol{y_i}$ is conditioned on context vector $\boldsymbol{c_i}$ calculated by,

$$\boldsymbol{c_i} = \sum_{j=1}^{len(\boldsymbol{x})} \alpha_{ij}\boldsymbol{h_j}, \tag{7}$$

where α_{ij} is the weight for hidden state $\boldsymbol{h_j}$ to context vector $\boldsymbol{c_i}$.

Table 1: Composition of our Japanese-Indonesian Parallel Corpus

Name of Source	# of pairs	portion
Open Subtitle 2016	695,582	95.87%
Asian Language Treebank	20,106	2.77%
Tanzil	8,127	1.12%
Global Voices	1,208	0.17%
Tatoeba	472	0.07%
Total	725,495	100%

Table 2: Statistics of training, development, and test sets

data division	# of pairs	portion
test set	72,552	10%
development set	65,295	9%
training set	587,648	81%
Total	725,495	100%

3 Experiment Settings

3.1 Dataset

There is no standard parallel corpus for Japanese-Indonesian machine translation research as we mentioned. Thus we collect various parallel data, as summarized in Table 1.

The parallel corpus of highest quality is *Asian Language Treebank*[*2] but consists of around 20,000 pairs only. Therefore, we decided to add subtitles and other parallel data to our corpus. All the other parallel corpora can be downloaded from OPUS[*3].

The limitation of our dataset is that the majority of the subtitles are incomplete sentences with considerable amount of noise. The characteristics and preprocesses of each parallel corpus to remove the noise are described below.

[*2] http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/

[*3] http://opus.lingfil.uu.se/

***Asian Language Treebank* [10]** The treebank is created by a joint project among seven institutes, BPPT, I2R, IOIT, NECTEC, NIPTICT, UCSY, and NICT, for making parallel treebank for eight languages. The original sentences were samples of 20,000 sentences from English Wikinews, and these sentences were translated into the other seven languages.

Global Voices The corpus contains news articles from the website of globalvoices.org. It was originally 1281 sentences, but we reduced to 1208 sentences due to bad translations. Some translations were not in the correct order and sometimes in other languages, such as Japanese and Arabic. We filtered them out manually.

Open Subtitle 2016 This corpus is from movie subtitles and mainly occupies our dataset. The original data is quite noisy as it contains song lyrics, shifted translations, and rarely used symbols. It also contains many unbalanced parallel data, which means the amount of information in source and target sentences are unequal. We reduced the original 729,162 sentences to 695,582 sentences. We removed data noise using regular expressions as well as manual check of samples.

Tanzil A collection of Quran translations compiled by the Tanzil project. Because of different interpretation and explanation in each language, the Indonesian version tends to contain more information and sometimes much longer than the Japanese version. We calculate the ratio of Japanese to Indonesian sentence and only allow sentence pairs with ratio bigger than 0.8. The corpus size is reduced from 18708 to 8127 sentences.

Tatoeba A corpus from tatoeba.org, a free collaborative online database of example sentences for foreign language learners. It contains 472 simple sentences.

After preprocessing the data, we divide them into test set, development set, and training set as shown in Table 2. We kept the ratio of sentence pairs from each data source consistent among training, development, and test set.

Finally, we use *MeCab*[8] as the Japanese tokenizer and english tokenizer provided in *Moses*[6] for the Indonesian as it has similar syntax to English.

3.2 Implementations

We evaluate the performance of Japanese-to-Indonesian machine translation on:

- *Moses*: phrase based SMT

Table 3: Setting of the NMT models. Layer size (l. size), word embedding dimension (embed. size), and attention mechanism (attn).

Name	layers	l. size	embed. size	attn
RNNenc	1	1024	1024	no
mRNNa	3	1024	1024	yes
biRNN	1	1000	620	yes

- *RNNenc* [11]: simple RNN encoder-decoder model
- *mRNNa* [11]: multi-layer RNN with attention
- *biRNN* [1]: bidirectional RNN

We use GRUs for all the NMT models and set all the vocabulary size to 60,000 of each languange. All NMT models are trained for 10 epochs. Other settings are shown in Table 3.

On *Moses*, we use GIZA++ for word alignment. Tokenization, truecasing, and cleaning are done before training. In the cleaning process, we limit sentence length to 80. The system uses a lexicalized reordering model (setting *msd-bidirectional-fe* in *Moses*) and 3-gram language model. Then, the system is tuned with our development set.

We use implementations of *RNNenc* and *mRNNa* available at *Tensorflow* library[*4]. On *RNNenc*, the embedding attention is removed and the model is trained with only one layer. Buckets of both models are set to 4 that fit sentence with length at most 50. We also changed *SGD* to *Adagrad optimizer* for its superior optimization capability.

As for *biRNN*, we use an implementation by the authors[*5]. It has the same parameter with model named *RNNsearch-50* in [1]. The beam-size for decoding is set to 12.

3.3 Evaluation Metric

Trained models are evaluated using BLEU with a script (*multi-bleu.perl*) provided in Moses. BLEU score enables us to compare different translation systems under the common corpus. Note that there is only one reference translation per source sentence in our test set.

4 Results and Discussions

Table 4 shows the BLEU scores on the test set, where *Moses* achieves the highest BLEU score:

[*4] https://github.com/tensorflow/models/tree/master/tutorials/rnn/translate

[*5] https://github.com/sebastien-j/LV_groundhog

Table 4: BLEU scores on the test set. The second and third column show the scores on translated sentences, and on the sentences removed unknown words, respectively.

system	All	w/o UNK
Moses	8.78	9.34
RNNenc	4.45	4.96
mRNNa	4.57	5.16
biRNN	4.85	6.45

8.78 on original translations and 9.34 on translations removed unknown words, respectively. *biRNN* achieves the highest score among the NMT models. Its BLEU score largely improves from 4.85 to 6.45 when removing unknown word. It shows that solving the unknown-words problem on NMT will give improvement on this model. Thus, we plan to apply byte pair encoding in future following the procedure in recent NMT systems.

RNNenc and *mRNNa* are largely different in their mechanism as *mRNNa* has attention model as well as 2 more layers in its network. However, improvement from *RNNenc* to *mRNNa* is limited, only 0.12 BLEU score on original translations. It may be due to the characteristic of our dataset that contains many incomplete and short sentences from subtitles. Such short sentences may not need the attention mechanism.

As we mentioned that *Open Subtitle 2016* contains a large number of unbalanced translations. It may disturb NMT to learn correct translations. We expect NMT models show higher performance if we could prepare a cleaner parallel corpus. In addition, the characteristics and differences on Japanese and Indonesian: affixes and grammars also have to be considered, which is our future work.

5 Conclusion and Future Work

This study has evaluated the performance of state-of-the-art machine translation systems on Japanese-to-Indonesian translation task. Although we collected as many parallel corpora as possible, still the amount and quality of the parallel corpora are not satisfactory. Insufficient parallel data remains a big challenge on this language pair.

To reduce the number of vocabularies, a morphological analyzer for Indonesian is desired, as in this research we only use english tokenizer. In addition, we did not handle affixes that actually makes the vocabulary size large due to their combinations. Preprocesses to Japanese particles and time-based verb form, which does not exist in Indonesian, may further improve the translation quality. We will work on these challenges as the next step.

Acknowledgement

This work has been conducted under the collaborative research between NTT Communication Science Laboratories and Osaka University.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proc. of Int'l Conf. on Learning Representations*, 2015.

[2] K. Cho, B. Merrienboer, D. Bahdanau, and Y. Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In *Proc. of Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, (Oct. 2014).

[3] K. Cho, B. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proc. of EMNLP 2014*, pp. 1724–1734, (Oct. 2014).

[4] S. Hochreiter and J. Schmidhuber. Long Short-term Memory. *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, (1997).

[5] Simon HS and A. Purwarianti. Experiments on Indonesian-Japanese Statistical Machine Translation. In *Proc. of Int'l Conf. on Computational Intelligence and Cybernetics*, (Dec. 2013).

[6] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proc. of ACL 2007*, pp. 177–180, (June 2007).

[7] P. Koehn, F. Och, and D. Marcu. Statistical Phrase-based Translation. In *Proc. of HLT-NAACL 2003*, pp. 48–54, (May-June 2003).

[8] T. Kudo, K. Yamamoto, and Y. Matsumoto. Applying Conditional Random Fields to Japanese Morphological Analysis. In *Proc. of EMNLP*, pp. 230–237, (July 2004).

[9] A. Purwarianti, M. Tsuchiya, and S. Nakagawa. Indonesian-Japanese Transitive Translation using English for CLIR. *Journal of Natural Language Processing*, Vol. 14, pp. 95–123, (2007).

[10] H. Riza, M. Purwoadi, Gunarso, T. Uliniansyah, A. Ti, S. Aljunied, L. Mai, V. Thang, N. Thai, V. Chea, R. Sun, S. Sam, S. Seng, K. Soe, K. Nwet, M. Utiyama, and C. Ding. Introduction of the Asian Language Treebank. In *Proc. of Oriental COCOSDA 2016*, (Oct. 2016).

[11] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton. Grammar as a Foreign Language. In *Proc. of NIPS 2015*, (Dec. 2015).