
Japanese-to-English Machine Translation Using Recurrent Neural Networks

Eric Greenstein
Stanford University
ecgreens@stanford.edu

Daniel Penner
Stanford University
dzpenner@stanford.edu

Abstract

Neural network machine translation systems have recently demonstrated encouraging results. We examine the performance of a recently proposed recurrent neural network model for machine translation on the task of Japanese-to-English translation. We observe that with relatively little training the model performs very well on a small hand-designed parallel corpus, and adapts to grammatical complexity with ease, given a small vocabulary. The success of this model on a small corpus warrants more investigation of its performance on a larger corpus.

1 Introduction

Among the major problems in natural language processing, the problem of machine translation has proved both one of the most enticing, as well as one of the least approachable. Over the course of its history many approaches have been applied, from traditional, labor-intensive rule-based methods to the more recent statistical methods. Still, as a couple minutes spent on Google Translate, an online translator which uses statistical machine translation, will indicate, there is still a long way to go before one can consider this problem solved in any useful capacity.

However, the efficacy of a machine translation system is also heavily dependent on the language pair under consideration. For example, though there are still grammatical structures which are not translated appropriately, statistical machine translation between language pairs such as French and English is considered to have achieved enough accuracy to be somewhat useful in practice. The same can be said of statistical machine translation between the majority of Romance languages, which in general produces substantially better results than machine translation between English and languages of non-European origin, such as Japanese.

Although the common root of these languages provides an explanation as to why they work better, another reason is the abundance of expert-translated corpora between English and the Romance languages, particularly the European Union parliamentary notes, which are simultaneously recorded in all the official languages of the participating nations.

Recent advances in deep learning have led to the dominance of neural network-based methods in various subfields of artificial intelligence, the success found in computer vision using convolutional neural network models. Though neural network-based machine translation models have yet to match the state-of-the-art phrase-based statistical learning methods, the gap is closing at an encouraging pace as new models tailored to the task of machine translation are being developed and fine-tuned [23].

In this paper we examine the performance of some recently developed models for machine translation using deep learning in application to Japanese-to-English machine translation.

2 Related Work

Machine translation has been an active research topic since the 1950's [11]. Originally, systems were developed using dictionaries and rules for producing correct word order, and researchers tried to use knowledge of language to improve their models. In the 1990's, statistical methods based on corpora of translation examples began to emerge [13]. Eventually, these methods became dominant due to the availability of large corpora, software for performing basic translation processes (such as alignment, filtering, reordering, etc.), and computational speed. Some rule-based pieces do remain in machine translation systems, however.

Neural networks have also been applied to natural language processing for some time [15]. Using neural networks to learn a statistical model of the distribution of word sequences, and operating at a large scale, was achieved in 2003 by Bengio et al. [4]. Recurrent and recursive neural networks have been used successfully for natural language processing tasks and achieved close to state-of-the-art accuracy in machine translation [1] [2] [5] [14] [17] [21].

The particular problem of machine translation between English and Japanese has a long history as well. In a 1982 paper by Nagao [18] implements a rule-based machine translation system by attempting to transfer grammatical concepts between the two languages. More recently, a paper by Tamura, et al. [24] applies statistical machine translation methods to word alignment models using recurrent neural networks. However, the authors state that the results on machine translation achieve only a baseline level of success.

Recently, neural networks have received more attention in machine translation [12] [7] [23]. These models often take an encoder-decoder approach to learn translations. In this approach, an encoder neural network reads a source sentence and encodes it into a fixed-length vector. A translation is then made by the decoder, which decodes the fixed-length vector into a sentence of variable length. The system is trained to maximize the conditional probability of a correct translation given a source sentence. Recurrent neural networks with long short-term memory (LSTM) or gated recurrent units (GRUs) achieve close to state-of-the-art performance to conventional phrase-based systems on some translation tasks [23].

3 Approach

In this paper we examine the performance of two recently developed models for neural machine translation on a handful of parallel corpora.

3.1 Model

Our main experiments were run using the model proposed in a 2014 paper by Bahdanau, Cho, and Bengio [3], which the authors call RNNsearch. RNNsearch is a generalization of a previous model proposed earlier in 2014 by Cho et al. [7] called RNN Encoder-Decoder, which is an architecture that learns to simultaneously align and translate. The implementation of these two models that we trained and adapted is available in a public Python framework on Theano called Groundhog, developed by Pascanu, Gulchere, and Cho from the LISA lab at University of Montreal.¹

A bidirectional recurrent neural network (BiRNN) is used as the encoder in this architecture. BiRNNs, first proposed in 1997 by Schuster and Paliwal [22], concatenate the hidden state given by the forward hidden state of an RNN that reads the source sentence as it is ordered and the backward hidden state of an RNN that reads the source sentence in reverse. These models are able to capture summaries of both the preceding and following words around a target word. For the activation functions of the RNN, a gated hidden unit (GRU), proposed by Cho et al. in 2014 [7] are used. GRUs are able to learn long-term dependencies in data, which is important in machine translation. These units are similar to LSTM units.

Specifically, the forward hidden states are computed as follows:

$$\vec{h}_i = \begin{cases} (1 - \vec{z}_i) \circ \vec{h}_{i-1} + \vec{z}_i \circ \vec{h}_i & i > 0 \\ 0 & i = 0, \end{cases}$$

¹Available at <https://github.com/lisa-groundhog/GroundHog>.

where

$$\vec{h}_i = \tanh(\vec{W}\vec{E}x_i + \vec{U}[\vec{r}_i \circ \vec{h}_{i-1}])$$

is the i^{th} hidden state,

$$\vec{z}_i = \sigma(\vec{W}_z\vec{E}x_i + \vec{U}_z\vec{h}_{i-1})$$

is the i^{th} update gate, and

$$\vec{r}_i = \sigma(\vec{W}_r\vec{E}x_i + \vec{U}_r\vec{h}_{i-1})$$

is the i^{th} reset gate, and the backward hidden states are computed in the same fashion and then concatenated. In computing the forward and backward reset gate, hidden state, and update gates at each step i , the backward and forward gates use different weight matrices $\vec{W}_z, \vec{W}_r, \vec{W} \in \mathbb{R}^{n \times m}$, $\vec{U}_z, \vec{U}_r, \vec{U} \in \mathbb{R}^{n \times n}$, but share the same word embedding matrix $\vec{E} \in \mathbb{R}^{m \times K_x}$. m and n are the word embedding dimensionality and the number of hidden units.

A RNN with GRUs is also used as the decoder. The context vectors are recomputed at each step using an alignment model. With the context vector fixed as the final forward hidden state, the RNNsearch model reduces to the RNN Encoder-Decoder model mentioned above. In the decoder the hidden states s_i are computed as:

$$s_i = (1 - z_i) \circ s_{i-1} + z_i + \tilde{s}_i,$$

using raw states

$$\tilde{s}_i = \tanh(Ey_i + U[r_i \circ s_{i-1}] + Cc_i),$$

and update gates

$$z_i = \sigma(W_zEy_i + U_zs_{i-1} + C_zc_i),$$

where

$$r_i = \sigma(W_rEy_i + U_rs_{i-1} + C_rc_i)$$

are the reset gates. E is the word embedding matrix in the target language, $W_z, W_r, W \in \mathbb{R}^{n \times m}$, $U_z, U_r, U \in \mathbb{R}^{n \times n}$, and $C_z, C_r, C \in \mathbb{R}^{n \times 2n}$ are weights. The initial hidden state s_0 is computed by $s_0 = \tan(W_s \vec{h}_1)$, where $W_s \in \mathbb{R}^{n \times n}$.

c_i are the context vectors, recomputed at each step as

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j,$$

where T_x is the length of the source sentence and the weights α_{ij} define the alignment model as follows:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

which is a normalization of

$$e_{ij} = v_a^T \tanh(W_a s_{i-1} + U_a h_j)$$

which model the alignment of input word i with output word j .

The probability of a target word y_i is given by:

$$p(y_i | s_i, y_{i-1}, c_i) \propto \exp(y_i^T W_o t_i)$$

where

$$t_i = [\max\{\tilde{t}_{i,2j-1}, \tilde{t}_{i,2j}\}]_{j=1, \dots, l}^T$$

and $\tilde{t}_{i,k}$ is the k -th element of a vector \tilde{t}_i , which is computed by

$$\tilde{t}_i = U_o s_{i-1} + V_o E y_{i-1} + C_o c_i$$

$W_o \in \mathbb{R}^{K_y \times l}$, $U_o \in \mathbb{R}^{2l \times n}$, $V_o \in \mathbb{R}^{2l \times m}$, and $C_o \in \mathbb{R}^{2l \times 2n}$ are weight matrices.

3.2 Data

There are a number of large English-Japanese parallel corpora publicly available. Among them is the roughly 500,000 sentence-pair Kyoto wiki corpus, consisting of translated paragraphs written about various aspects of life and culture in Kyoto, Japan. Other large corpora include the TED corpus, a large collection of bilingual subtitles from TED talks, and the Tanaka corpus, a roughly 150,000 sentence-pair collection of student translated Japanese sentences. The first two translations are professional, while the third consists mostly of accurate translations, with occasional unnatural English translations of more grammatically complex or idiomatic phrases.

We use a publicly available Japanese language parser called TinySegmenter (available at [9]) to split the sentences of the corpora into tokens (roughly the equivalent of words in English – the distinction is substantially more ambiguous in Japanese).

Due to the time-intensive training required of these translation models on large corpora, we found that training on any of the corpora listed above would not be possible with our limited resources and time. At first we decided to train on a subset of the above corpora, but found that by randomly selecting a subset of the Tanaka corpus, the variety in sentence structure and vocabulary (compounded with the inconsistency of the translations and transcriptions – for example, there are multiple ways to write most words in Japanese) proved prohibitory to training a good model on a small dataset.

This led us to develop our own hand-crafted parallel corpora to explore how quickly the models can adapt to various linguistic features being introduced into a small corpus with a small vocabulary. The features we wished to include in the corpus were (a) a relatively small vocabulary compared to the number of sentences, (b) consistent transcriptions of words and consistent word segmentation, (c) consistent translation of grammatical phrases, and (d) a variety of sentences of different grammatical structures.

The hand-designed corpus we settled on, a subset of which is shown in Figure 1 below, consists of simple sentences of similar forms such as “He is going to school,” varying the tense (“He went to school”), the subject (“She went to school”), the indirect object (“He is going to the bank”), and adding negation (“He is not going to school”). The idea is that if we restrict the vocabulary as much as possible while still varying the sentence structure in subtle, but important ways, we can check on slight out-of-sample variations whether or not the model can extrapolate to sentences constructed from grammatical structures and vocabulary that it has learned.

She is going to the store .	彼女は 店 に 行く 。
She is going to the bank .	彼女は 銀行 に 行く 。
She is going to the street .	彼女は 道 に 行く 。
He is going to the concert .	彼は コンサート に 行く 。
He is going home .	彼は 家 に 行く 。
He is going to the bus stop .	彼は バス停 に 行く 。
He is going to school .	彼は 学校 に 行く 。
She is my friend .	彼女は 友達 だ 。
She is my teacher .	彼女は 先生 だ 。
She is my professor .	彼女は 教授 だ 。
He is my boss .	彼は 上司 だ 。
He is my enemy .	彼は 敵 だ 。
She is not going to the party .	彼女は パーティー に 行か ない 。
She is not going to the club .	彼女は クラブ に 行か ない 。
He is not going to the bathroom .	彼は トイレ に 行か ない 。
He is not my professor .	彼は 教授 じゃ ない 。
He is not my teacher .	彼は 先生 じゃ ない 。
He is not my friend .	彼は 友達 じゃ ない 。
She is not my enemy .	彼女は 敵 じゃ ない 。
She is not my boss .	彼女は 上司 じゃ ない 。
He went to the store .	彼は 店 に 行っ た 。
He went to the bank .	彼は 銀行 に 行っ た 。
He went to the street .	彼は 道 に 行っ た 。
She went to the concert .	彼女は コンサート に 行っ た 。
She went home .	彼女は 家 に 行っ た 。
She went to the bus stop .	彼女は バス停 に 行っ た 。
She went to school .	彼女は 学校 に 行っ た 。
He was my friend .	彼は 友達 だ 。
He was my teacher .	彼は 先生 だ 。
He was my professor .	彼は 教授 だ 。

Figure 1: Subset of handmade corpus.

3.3 Measuring the Results

We evaluate the performance of our models on the corpora by use of a standard evaluation metric for machine translation called BLEU. This metric was initially proposed in a 2002 paper by Papineni, Kishore, et al. [19]. It measures precision of a machine translated phrase in comparison to a human translated reference phrase by counting matching n -gram pairs and taking its proportion to

the number of words in the reference phrase. Papineni, Kishore, et al. argue that the BLEU metric strongly correlates with expert human evaluation, and thus in general makes for a good substitute evaluation metric, in the absence of the extremely time-intensive method of human evaluation. The BLEU score gives us a means to compare the results from our two models to one another, for a fixed corpus.

4 Experiment

In this section we detail the experiments we ran, using the RNNsearch model described in section 3.1, and the results we obtained.

4.1 Tanaka Corpus

We first attempted to train an RNNsearch model on a subset of the publicly available, approximately 150,000 sentence-pair, student-translated Tanaka corpus. For this model, we set the size of the hidden layer to be 1000 units, the word embedding dimensionality to be 620 and the size of the maxout hidden layer in the deep output to be 500, and the number of hidden units in the alignment model to be 1000. We initialized the recurrent weight matrices as random orthogonal matrices. W_a and U_a were initialized by sampling each element from the Gaussian distribution of mean 0 and variance 0.001. All the elements of V_a and all the bias vectors were initialized to zero. Any other weight matrix was initialized by sampling from the Gaussian distribution of mean 0 and variance 0.01. Training was performed using minibatch stochastic gradient descent using Adadelta (Zeiler 2012) to update the learning rate.

However, after training on a subsample of 1000 sentence-pairs for many hours, we found that the training examples were still being predicted very badly, and the loss function was decreasing at too slow a pace for us to obtain any results by the deadline. We found that some training examples were predicted exactly (indicating that rather than learning any sort of structure it was just fitting examples to one another), while others were simply guessed wrong. Figure 2 below shows a couple examples of translations given by the model:

Input: UNK が ひろし に 試験 を UNK ている 。 <eol>	Input: 時々、私の犬は UNK の間に吠えます。 <eol>
Target: the yakuza were tormenting hiroshi . <eol>	Target: sometimes my dog barks in the middle of the night . <eol>
Input: UNK が ひろし に 試験 を UNK ている 。 <eol>	Input: 時々、私の犬は UNK の間に吠えます。 <eol>
Output: the yakuza were tormenting hiroshi . <eol>	Output: i m so drunk now that i m seeing two keyboards . <eol>

Figure 2: Sample training set translations, Tanaka corpus.

For these reasons we switched over to designing a small hand-crafted parallel corpus which we could train quickly and obtain accurate out-of-sample translations.

4.2 Hand-Crafted Corpus

In the absence of any conclusive results from training a model on the Tanaka corpus, we resolved to design our own hand-crafted parallel corpus, as detailed in section 3.2, and to train an RNNsearch model on it. Using a test set we designed by taking examples from the training set and permuting the vocabulary slightly, so as to create different but very similar examples, we were able to test our model's predictions, to see if it could extrapolate from grammatical structures and vocabulary that it has seen during training. We found that the model converged to near-perfect accuracy on the training set within minutes, unsurprising due to the size of the dataset, vocabulary, and model. A few translations made by our model are shown in Figure 3 below.

Using our test set as described above as a validation set to decide the best hidden layer size (since we definitely had to shrink it from the previous, much larger model to avoid overfitting), we trained models and computed the BLEU scores of our translations. The best score was obtained with a hidden layer size of 10, for which we obtained a score of 0.73. Out of the 32 examples in the test set, 21 of our translations were exactly correct, and another 8 generated the correct translation and marked it as second best (the model outputs 24 translations in order of likelihood, as in Figure 4 below, where the correct translation is marked in red)

```

Input: 彼女は敵だった。 <eol>
Target: She was my enemy . <eol>
Input: 彼女は敵だった。 <eol>
Output: She was my enemy . <eol>

Input: 彼は家に行く。 <eol>
Target: He is going home . <eol>
Input: 彼は家に行く。 <eol>
Output: He is going home . <eol>

Input: 彼は先生だった。 <eol>
Target: He was my teacher . <eol>
Input: 彼は先生だった。 <eol>
Output: He was my teacher . <eol>

```

Figure 3: Sample training set translations, hand-crafted corpus.

```

Parsed Input: 彼女はコンサートに行く。 <eos>
0.00663540713026: She is going to the concert .
5.27218884914: She is my professor .
6.86046250971: She is UNK .
9.47003285695: She is going is not my professor .
9.67765936555: She is going to the He .
Input Sequence: 彼女は家に行く。
How many samples? 5
Parsed Input: 彼女は家に行く。 <eos>
0.291400269567: She is going to school .
1.76811346484: She is going home .
2.53733918163: She is my friend .
6.52093583348: She is my boss .
7.40240003171: She is UNK .

```

Figure 4: Sample test set translations. The correct translations are marked in red.

5 Conclusion

We successfully implemented the RNNsearch model and trained it on different datasets. Our model was able to extrapolate to out-of-sample sentences of similar structure and vocabulary to examples in the training set, making exact translations with relatively high accuracy. Although we did not have the time or resources to train on a larger corpus, the result is encouraging that this model, if trained on a larger dataset, can yield good predictions as well.

6 Acknowledgements

We would like to thank Professor Socher and the teaching staff for their assistance this quarter.

References

- [1] Allauzen, Alexandre, et al. "LIMSI@ WMT11." Proceedings of the Sixth Workshop on Statistical Machine Translation. Association for Computational Linguistics, 2011.
- [2] Auli, Michael, et al. "Joint Language and Translation Modeling with Recurrent Neural Networks." *EMNLP*. 2013.
- [3] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
- [4] Bengio, Yoshua, et al. "A neural probabilistic language model." *The Journal of Machine Learning Research* 3 (2003): 1137-1155.
- [5] Bengio, Yoshua, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.8 (2013): 1798-1828.
- [6] Brockett, Chris, et al. "English-Japanese example-based machine translation using abstract linguistic representations." Proceedings of the 2002 COLING workshop on Machine translation in Asia-Volume 16. Association for Computational Linguistics, 2002.
- [7] Cho, Kyunghyun, et al. "Learning phrase representations using rnn encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078(2014).
- [8] Forcada, Mikel L., and Ramn P. eco. "Recursive hetero-associative memories for translation." *Biological and Artificial Computation: From Neuroscience to Technology*. Springer Berlin Heidelberg, 1997. 453-462.

- [9] Hagiwara, Masato. TinySegmenter in Python. <http://lilyx.net/tinysegmenter-in-python/>
- [10] Hochreiter, Sepp, and Jrgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.
- [11] Hutchins, John. "The history of machine translation in a nutshell." Retrieved December 20 (2005): 2009.
- [12] Kalchbrenner, Nal, and Phil Blunsom. "Recurrent Continuous Translation Models." *EMNLP*. 2013.
- [13] Koehn, Philipp. Statistical machine translation. Cambridge University Press, 2009.
- [14] Le, Hai-Son, et al. "LIMSI@ WMT'12." Proceedings of the Seventh Workshop on Statistical Machine Translation. Association for Computational Linguistics, 2012.
- [15] R. Miikkulainen and M.G. Dyer. Natural language processing with modular neural networks and distributed lexicon. *Cognitive Science*, 15:343-399, 1991.
- [16] Mikolov, Tomas, et al. "Extensions of recurrent neural network language model." Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE, 2011.
- [17] Mikolov, Tomas, et al. "Recurrent neural network based language model." INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010. 2010.
- [18] Nagao, M. 1984. A framework of a mechanical translation between Japanese and English by analogy principle. In A. Elithorn. and R. Bannerji (eds.) *Artificial and Human Intelligence*. Nato Publications. pp. 181-207.
- [19] Papineni, Kishore, et al. "BLEU: a method for automatic evaluation of machine translation." *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002.
- [20] Schmidhuber, Jrgen. "Deep learning in neural networks: An overview." *Neural Networks* 61 (2015): 85-117.
- [21] Schwenk, Holger, Anthony Rousseau, and Mohammed Attik. "Large, pruned or continuous space language models on a gpu for statistical machine translation." Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT. Association for Computational Linguistics, 2012.
- [22] Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45(11), 2673-2681.
- [23] Sutskever, Ilya, Oriol Vinyals, and Quoc VV Le. "Sequence to sequence learning with neural networks." *Advances in Neural Information Processing Systems*. 2014.
- [24] Tamura, Akihiro, Taro Watanabe, and Eiichiro Sumita. "Recurrent neural networks for word alignment model." Proc. ACL. 2014.