Japanese to English Machine Translation using Preordering and Compositional Distributed Semantics

Sho Hoshino Hubert Soyer Yusuke Miyao Akiko Aizawa
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda, Tokyo, Japan
{hoshino,soyer,yusuke,aizawa}@nii.ac.jp

Abstract

The pipeline of modern statistical machine translation (SMT) systems consists of several stages, presenting interesting opportunities to tune it towards improved performance on distant language pairs like Japanese and English. We explore modifications to several parts of this pipeline. We include a preordering method in the preprocessing stage, a neural network based model in the tuning stage and a recurrent neural network language model in the postprocessing stage.

To our knowledge this is the first work tightly integrating a neural network based model into the tuning stage of a SMT system for the Japanese-English language pair. As a first step in this direction we provide several insights into how this integration should be approached and give rise to future work in this area.

1 Introduction

Modern models for statistical machine translation constitute a pipeline of different components. This pipeline usually involves a preprocessing part, a language model, a translation model and a postprocessing part. While this or a similar structure is basis of most systems and generally agreed upon, a lot of research has been focusing on modifying and extending the individual components.

Many parts rely on probabilities acquired through word frequencies in fixed contexts, discarding additional syntactical and semantical information. Problems arising from these strong assumptions become particularly apparent when dealing with distant language pairs like Japanese and English.

We focus exclusively on translating Japanese sentences to English and take several measures to inject additional information into the pipeline of our baseline system.

- We apply preordering to the input text as a way of compensating for syntactic differences between English and Japanese.
- We insert scores into the translation model of our baseline system that are computed from semantically meaningful distributed vector representations.
- As postprocessing, we utilize a recurrent neural network language model to re-score the 100 best translation candidates for each output sentence of our system. Being able to handle variable length context, it complements the n-gram based language model used within the pipeline.

2 System

We built our baseline system with Moses (Koehn et al., 2007) as a phrase-based machine translation system loosely following the setup described by the WAT 2014 organizers (Nakazawa et al., 2014), with some modifications.

We will quickly go through every step of our training.

1. Tokenization (section 3)
2. Training of compositional distributed vector representations (section 2.1.2)
3. Preordering (section 2.1.1)
4. Generation of 6-gram language model with SRILM
5. Training of translation model with MGIZA++
6. Tuning with compositional distributed semantics features
7. Translation of devtest and test set using Moses decoder
8. Training of RNN LM and reranking of 100 best translation candidates for every sentence

Proceedings of the 1st Workshop on Asian Translation (WAT2014), pages 55–63,
Tokyo, Japan, 4th October 2014.

9. Evaluation of final output through BLEU and RIBES scores and submission to WAT 2014 human evaluation

2.1 Extensions

In the following, we will describe the modifications to our baseline system. They are grouped into three parts, one part describing the preordering of the Japanese input text, the second part explaining the integration of a neural network based distributed compositional semantics model and lastly, the reranking of the translation candidates for each sentence using a recurrent neural network language model.

2.1.1 Preordering Japanese Text

We employ a preordering method for Japanese-to-English translation. We preorder the input text in the preprocessing stage to reduce the difference between the word order of the Japanese input sentence and the word order of the English target sentence.

Parts of a Japanese sentence can be scrambled/shuffled without changing the meaning of the sentence or making it grammatically incorrect. Therefore, one English sentence can potentially correspond to several shuffled variations of the same Japanese sentence.

Yoshida et al. (2014) show that normalizing the word order of Japanese sentences can benefit readability.

Taking scrambling into account not only increases readability, it also plays an important role in machine translation (Isozaki et al., 2014). Hayashi et al. (2013) were able to improve the results of statistical machine translation systems by generating English determiners in the Japanese input text and reordering its words. Kudo et al. (2014) applied preordering and generated zero-subjects to achieve state-of-the-art results on a web-text corpus.

We employ the preordering rules introduced in (Hoshino et al., 2014) to reduce order ambiguity in the Japanese input text. This method has achieved state-of-the-art results on the NTCIR patent corpus.

Following this method, we parse input sentences with the Japanese dependency parser KNP to obtain chunked[1] dependency and coordination labels corresponding to the dependency. After that, we apply the following three rules: Rule 1 transforms each chunked sentence into a form that is more suitable for the preordering rules that follow this step. Rule 2 reorders entire chunks while Rule 3 reorders the words inside of every chunk resulting in the final reordered version of the sentence.

Rule 1 (chunking) Given an input sentence with l chunks ($input = c_1\ c_2\ \ldots\ c_l$) we merge all coordinated chunks into one chunk ($c_1\ c_2 \ldots c_l \rightarrow c_1\ c_2 \ldots c_m$ where $m \leq l$). After that, we split punctuations[2] from chunks applying the following rules:

Rule 1-A A chunk $c_x(1 < x \leq m)$ will not be split when a predecessor chunk c_{x-1} is a noun.

Rule 1-B Even if Rule 1-A is applied to a chunk c_x, the chunk will be split into three new chunks (words,particle,punctuation) if this chunk ends with a particle[3] .

Rule 1-C A chunk c_x will always be split into words and punctuation when it ends with a punctuation.

Rule 2 (inter-chunk reordering) After Rule 1 produced a new chunked sentence, all of the n chunks between punctuations ($c_1\ c_2\ \ldots\ c_n$ where $c_x(1 \leq x \leq n)$ is not a punctuation) are reordered ($c_1\ c_2\ \ldots\ c_n \rightarrow w1 \ldots w_{q-1}\ c_{i-1} \ldots c_1\ w_q\ c_n \ldots c_{i+1}$), where a chunk $c_i(1 \leq i \leq n)$ contains q words, ending with a particle word ($c_i = w_1, ..., w_{q-1}w_q$ where w_q is a particle).

Rule 3 (intra-chunk reordering) Given a chunk c_x which has p content words and q function words ($c_x = w_1\ \ldots\ w_p\ w_{p+1}\ \ldots\ w_{p+q}$), we swap the content words and the function words, then reverse the function words ($c_x = w_1\ \ldots\ w_p\ w_{p+1}\ \ldots\ w_{p+q} \rightarrow w_{p+q} \ldots w_{p+1}\ w_1\ \ldots\ w_p$).

Figure 1 illustrates how the introduced preordering rules are applied to an example sentence.
The sentence comprises 6 chunks two of which are labeled as coordinated by the KNP parser (表1と and 図2は).
Step 1 displays the basic, unaltered source sentence. In the second step the two coordinated

[1] This chunk actually is a bunsetsu, a linguistic unit in Japanese, as used in (Yoshida et al., 2014)

[2] We only regard Japanese comma "、" or period "。" as punctuation.

[3] We only consider the Japanese topic case marker particles "ga" and "ha".

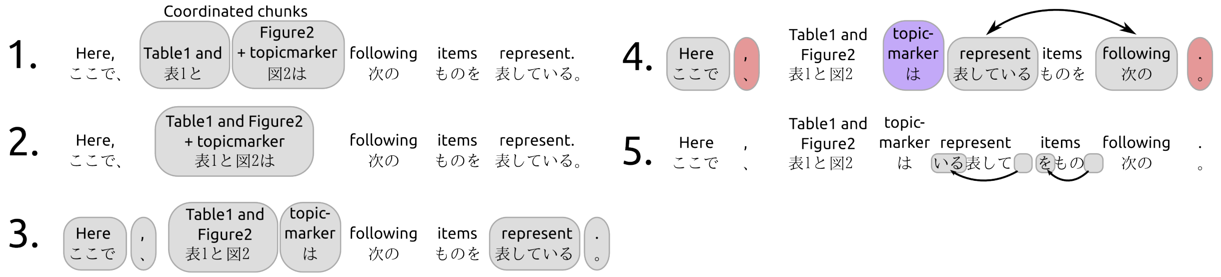

Figure 1: Illustration of the pre-ordering rules described in 2.1.1 to an example.

chunks are merged into one following Rule 1. Rule 1, Rule 1 A, B and C result in several splits (topic marker,“.”, “,”) in step 3.
In step 4, Rule 2 causes “represent” and “following” to be switched. Finally, Rule 3 is applied in step 5 and affects only the Japanese text. いる and を, both labeled as function words by the parser.

2.1.2 Compositional Distributed Semantics

Many machine learning algorithms require fixed length vectors as input. Of the various different ways to map words to vectors, neural network based models have proven very effective recently.

Vector representations (embeddings) created by these models have been utilized as features to achieve state-of-the-art results in several different Natural Language Processing tasks (Collobert et al., 2011; Mikolov et al., 2013; Baroni et al., 2014), giving rise to many new variants. Methods have been developed that are capable of embedding words from different languages into the same vector space (Zou et al., 2013), there are models that can induce representations of whole phrases or sentences instead of only single tokens (Socher et al., 2010; Le and Mikolov, 2014; Blacoe and Lapata, 2012) and there are models implementing a mixture of these two ideas, embedding phrases from different languages into the same vector space (Hermann and Blunsom, 2014; Cho et al., 2014).

Work has been published about integrating embeddings and neural network models into the statistical machine translation pipeline for various language combinations, however, we are not aware of any previous work attempting this for the Japanese/English language pair.

As a first step in this direction, we integrate the model introduced by Hermann et al. (Hermann and Blunsom, 2014) into our baseline system, specifically, we use a slightly modified version of the “BI” model described in the paper.

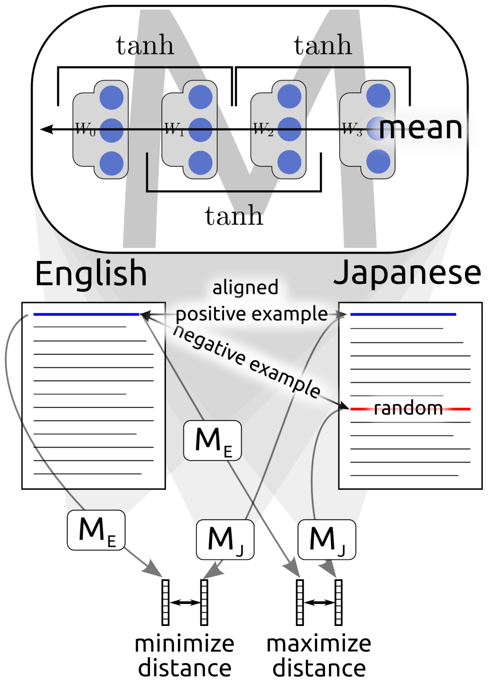

Figure 2: Illustration of model described in (Hermann and Blunsom, 2014), slightly modified to use mean instead of addition

From this point on we will refer to the model as embedding model or neural network model. Learned in an unsupervised way, we hope to indirectly inject context knowledge through the neural network model into our baseline machine translation system.

Figure 2 illustrates how the model is trained. The training is centered around a model M that looks up a word representation w_i for each word in a given sentence and combines them by wrapping a hyperbolic tangent function ($tanh$) around the sum of the vectors of each bi-gram and computing the mean of these intermediate results. The original version described in (Hermann and Blunsom, 2014) applies addition instead of the mean. However, in preliminary experiments on an information retrieval task we obtained more promising results using the mean and therefore decided to exploit this insight.

One training step requires this model to be applied to three sentences: One English sentence, the Japanese sentence aligned to this English sentence and a Japanese sentence chosen at random from the corpus. There are two sets of word representations, a Japanese set and an English set, that are initially initialized at random by sampling values from a Gaussian centered at 0 with a standard deviation of 0.1. When looking up the vector representations to compose in the model, we choose the set of representations corresponding to the language that the currently handled sentence is in.

During training we then modify the word vectors to minimize the Euclidean distance between the vectors composed for the sentences of the valid, sentence aligned Japanese-English sentence pair and maximize the distance between the vectors of the English sentence and the randomly sampled Japanese sentence. By repeating this procedure for many sentence pairs the model learns to distinguish between valid translation pairs and invalid, randomly sampled pairs. At application time we exploit this property and compute a normalized distance between two given phrases or sub-phrases to judge their similarity.

Because this model only weakly incorporates order information on the bi-gram level it is applicable to very short phrases and even single tokens and can potentially even deal with preordered text. Due to this flexibility we can insert it into almost any part of the machine translation pipeline.

We trained the model on the 1 million parallel sentences in ASPEC that feature the highest alignment scores. We omitted the remaining 2 million sentence pairs completely to avoid introducing errors caused by incorrect alignment. Replacing tokens that occur less than 6 times in the training set resulted in more stable learning and better performance on a simple information retrieval task. After hyperparameter optimization the vectors in our embedding model were chosen to be 128 dimensional, the margin for the noise-contrastive objective was set to 1.0 and we regularized the word representations with a $l2$-regularization of 10^{-5}. For faster training we utilized RMSProp (Tieleman and Hinton, 2012) with a decay of 0.99 and a learning rate of 0.001.

We explored two ways to include the method into our baseline system.

Feature Function We created a new feature function that is utilized in the tuning phase of the log-linear translation model of our baseline. Moses allows us to access the text of each sub-phrase candidate pair evaluated during decoding. For each sub-phrase we compute its corresponding Japanese and English vector representation using our embedding model. We proceed by calculating a similarity score between the English and the Japanese sub-phrase vector. We chose the angular similarity, a normalized version ($[0, 1]$) of the cosine similarity, as it was superior to the Euclidean similarity in our previous experiments on an information retrieval task. Depending on the outcome we create one of 3 *Sparse Features*. One feature for highly similar phrases (similarity $>$ 0.5), one for less similar phrases (similarity $\leq$ 0.5) and one for out-of-vocabulary cases where none of the tuples in one or both phrases could be found. This feature split is necessary to counteract problems with out-of-vocabulary cases. We encountered these cases specifically for phrases containing mostly named entities. Named entities are often not part of the pre-trained word representations. Even though moses would translate these tokens correctly (often just by transliteration) the score produced by the neural model would be not defined due to out-of-vocabulary errors. Not using this feature split lead to very poor performance.

Phrase Table We computed a score for each phrase tuple in the phrase table of our system, following (Cho et al., 2014). We iterated through all phrase combinations present in the phrase table and added an additional similarity score computed by the neural model in the same way as described above for the feature function case. This does not require extending our baseline system, the system only needs to incorporate one more feature when reading the phrase table.

2.1.3 Reranking with the Recurrent Neural Network Language Model

Neural Networks have been applied successfully in language modeling tasks over the past years (Mikolov et al., 2011a) and are increasingly often found in combination with or as an extension to statistical machine translation systems (Devlin et al., 2014; Cho et al., 2014; Zhang et al., 2014; Liu et al., 2014).

We employ the Recurrent Neural Network Language Model Toolkit (RNNLM) (Mikolov et al., 2011b) to rerank the 100 best translation candidates for each sentence and use the best candidates for our submission.

3 Data

We use only the provided ASPEC corpus and do not rely on any external resources. We train our language model with all 3 million sentences of the training set. For the translation model we only utilize the top 1 million sentences (regarding alignment scores). For hyperparameter tuning we use the development split of the corpus. Our preliminary evaluation is conducted on the *devtest* split, only the final results specified in the paper were calculated on the *test* split.

The default moses tokenization script does not perform Unicode NFKC normalization. Converting Japanese full width roman characters to half width characters is crucial to tackle transliteration cases from Japanese to English. Therefore we apply the following preprocessing steps before executing the moses tokenization script.

1. Conversion of XML entity names to Unicode characters
2. Character normalization by Unicode NFKC and deletion of double spaces
3. First letter lowercasing in all sentences without an all-uppercase first word
4. Conversion of double quotations to “ and ”
5. Commas, periods, and round brackets were regarded as tokens

In the final submissions, we applied the Moses detokenization script and converted the first letter of each sentence to upper case in order to avoid possible biases in the human evaluation.

4 Evaluation

We evaluated our results on the test split provided with the ASPEC corpus. To avoid tokenization mismatches we applied the same tokenization method to the test set that we had previously used on the training set and computed BLEU and RIBES scores with the Travatar[4] scoring script. The scores published on the WAT automatic evaluation website differ due to tokenization mismatches.

Preordering	Embeddings	BLEU	RIBES
No	None (Baseline)	19.16	63.41
No	Feature Function	18.95	63.48
No	Phrase Table (full)	18.82	63.23
Yes	Feature Function	18.92	61.76
Yes	None (Baseline)	18.55	61.44
Yes	Phrase Table (full)	-	-
No	Phrase Table (1 col)	14.53	60.59

Table 1: Scores for different ways to include the Compositional Distributed Semantics model, with and without preordering. In the *Feature Function* setting, the embedding model was integrated as a feature function into Moses; in the *Phrase Table* setting we employed the model to produce a score for each entry in the phrase table of the baseline translation model. *(full)* means that we used the default moses scores plus the newly computed score, for *(1 col)* we used only the score without the Moses default scores.

4.1 General Experiments

As listed in Table 1 our baseline achieved a score of 19.16 BLEU and 63.41 RIBES. Both of these scores can be considered low compared to results acquired e.g. on the Chinese to Japanese task of WAT 2014 where the phrase-based baseline model provided by the organizers achieved a BLEU score of 27.96. The same baseline system only achieved 18.45 on the Japanese to English task which confirms that the Japanese to English translation task is highly difficult.

Settings with preordering consistently perform worse than their counterparts without preordering, even though the employed preordering method has proven very successful (Hoshino et al., 2014) on the NTCIR patent corpus before. We suspect the main cause to be the different domain of the text. The NTCIR workshop revolved around text from the patent and legal domain while WAT 2014 is based on the ASPEC corpus comprising abstracts from scientific papers. Sentences in patent and legal text must be phrased in a very concise and unambiguous way and therefore feature a lot of recurring phrases. Writers of scientific text have more freedom to express content in English and can draw from a larger set of ways to phrase their ideas. This looser structure leads to mismatches when applying the heuristic preordering rules.

Both ways to integrate the scores computed with

[4] `http://www.phontron.com/travatar/evaluation.html`

the neural network model into the Moses pipeline perform roughly equally well, falling just a little short of the baseline but failing to beat it. (see *Phrase Table (full)* and *Feature Function*)

Examining phrase pairs and their scores revealed that the scores make sense in general, even identifying transliterations correctly with a high confidence. However, the model fails to capture an important property that prevents it from contributing information to Moses that is not already covered by the default features. It almost entirely neglects syntactic structure. Because of its objective to only distinguish valid translations from invalid ones, the model learns to ignore special characters, determiners and other words that do not possess a high discriminative value. The translation hypotheses of our baseline system, however, include many cases where the recognition of syntax and special characters is crucial to assign sensible scores. The values of word vectors of many stop words, determiners and special characters reveal that most of them are in close proximity to the zero vector. This vector consisting only of zeros constitutes the neutral element of the mean function as well as the hyperbolic tangent. Therefore, having or not having such a token in a sentence will only marginally change the composed representation.

This issue becomes particularly problematic for pairs of the form

Japanese	English
トークン	token
トークン	token !
トークン	the token the the

All three of these examples will have an almost identical score since the vectors of "!" and "the" were learned to be very close to the zero vector.

In an information retrieval context where processed sentences are sensible and sane, this property does not pose a huge problem. However, in the case of a phrase table or hypotheses in the tuning stage of an SMT system, cases like the one illustrated above occur regularly and should exhibit not the same but very different scores.

The *Phrase Table (1 col)* setting achieves 14.53 BLEU without utilizing the default Moses scores in the phrase table. We were surprised that having only one score value in the phrase table the system could reach a BLEU value that is not on an entirely different level than the baseline. We leave it to future research to investigate to what extend the neural model score can substitute the default Moses scores.

System	averaged Kendall's tau
Baseline	0.2990
Baseline + Preordering	0.3712

Table 2: Preordering Evaluation with Kendall's tau.

4.2 Preordering

To separate the evaluation of our preordering method from the machine translation evaluation, we calculated an intrinsic quality measure specific to preordering.

We apply the procedure previously introduced by (Isozaki et al., 2010b; Hoshino et al., 2014).

In our baseline method we rely on MGiza++ to align Japanese and English sub-phrases. Without preordering MGiza++ will perform a lot of non-monotonic alignments. The goal of preordering is to reduce the number of these non-monotonic alignments, in the best case leading to exclusively monotonic alignments. Utilizing Kendall's tau we can compare the alignments resulting from input with and without preordering. The closer this coefficient is to 1.0, the more monotonic are the alignments and the higher is the intrinsic benefit of the preordering. Table 2 lists the averaged coefficients computed with the baseline and the preordered input. It shows that our preordering performs better than the non-preordered system on this intrinsic measure.

Observations at the previously held NTCIR workshops indicated that better values in this measure correspond to better machine translation quality in terms of automatic as well as human evaluation. Our results in section 4.1 show that this intuition does not hold for the ASPEC corpus.

4.3 RNNLM Reranking

After training our baseline system including the neural network score feature function and obtaining 100 translation candidates for each input sentence of the test set, we apply the Recurrent Neural Network Language Model toolkit described in (Mikolov et al., 2011b) to re-rank these 100-best candidates according to their language model score.

Mikolov's implementation offers a variety of hyperparameters to tune the neural network training for optimal performance. Specifically, we

RIBES	BLEU	hidden	direct	Preordering
63.90	18.37	300	150	No
63.74	18.40	300	100	No
63.72	18.61	200	100	No
63.66	18.30	200	150	No
63.41	19.16	-	-	No
62.06	18.36	200	100	Yes
62.03	18.29	200	150	Yes
61.99	18.34	300	150	Yes
61.98	18.48	300	100	Yes
61.44	18.55	-	-	Yes

Table 3: Results for training the RNN LM with different hyperparameters applying it to re-rank the 100 best translation candidates for each output sentence of our SMT model. We varied the number of hidden units (*hidden*), the number of direct connections from input to output layer (*direct*) and kept the number of classes for the factored softmax fixed at 220 (following the recommendation in (Mikolov et al., 2011b) with $\#classes = \sqrt{\text{size of vocabulary}}$) and the number of steps of backpropagation-through-time fixed at 5. The *preordered* column indicates whether the preordering described in (Hoshino et al., 2014) was applied.

	BLEU	RIBES	human
Baseline	17.47	63.08	-5.750
Baseline + Preordering	17.01	61.08	-14.250

Table 4: Scores from WAT 2014 evaluation board.

adapted the number of hidden units and the number of direct connections from the input to the output layer. To choose the best settings we evaluated several configurations, the results for some of these configurations are displayed in Table 3. Reranking the translation candidates with the RNNLM improves the RIBES score while BLEU decreases instead. This pattern is consistent over all settings displayed in the table. Further, all settings with preordering perform worse than their counterparts without preordering. This goes along with previous observations presented in Table 1. The RIBES measure was specifically designed with distant language pairs like English and Japanese in mind(Isozaki et al., 2010a). With an increase of almost 0.5 from the non-preordered baseline to the best results with reranking and even more improvement for the preordered case, we conclude that reranking with an RNNLM improved our translation results significantly.

4.4 WAT 2014 Evaluation Board

The BLEU and RIBES scores on the official evaluation board as reported in Table 4 differ largely from the results presented in Table 1. This is due to tokenization mismatches.

Since our baseline system differs slightly from the system provided by the WAT 2014 organizers we submitted our baseline as a point of reference for our own experiments to the human evaluation. As our second submission to the human evaluation we chose our baseline with preordering. Preordering has proven successful in previous NTCIR workshops but has failed to improve upon the baseline in WAT 2014.

Table 4 shows that our baseline performs slightly worse than the baseline of the organizers. We are primarily interested in the difference between our baseline and our baseline with preordering. The human evaluation confirms that preordering has a negative effect when applied to the ASPEC corpus. According to the WAT 2014 homepage, the scores of the human evaluation are calculated by repeatedly comparing sentences from the submissions to their corresponding outputs of the baseline system provided by the organizers. They therefore indicate how a submission performs in comparison to the organizer's baseline. The fine grained results of the human evaluation in WAT 2014 offer a good opportunity to investigate the effects of our preordering method.

Examples illustrating the effect of the applied preordering method are shown in Table 5. We only picked examples featuring clear annotator agreement. -1 indicates that the organizers' baseline was assessed as superior to our system, 1 that it was evaluated to be inferior.

The biggest change in both of the examples shown in Table 5 was induced by the same preordering rule (Rule 2). Due to the lack of a topic marker in both source sentences Rule 2 caused the main verb to be brought to the front of the sentence.

Despite the similarity in the preprocessing step the quality of the translations differs heavily. While Example 2 was evaluated as superior to the organizers' baseline, Example 1 was received as just the opposite.

In example 1, the preordered main verb ("discussed") was omitted entirely in the output of our system, changing the meaning from a discussion to a statement.

On the contrary, the output of our system for ex-

Example 1: Negative Impact of Preordering	
human judgment	-1 -1 -1
Source	プラスチック以外の非金属製の義肢装具材料について論じた。
Preordered Source	論じた ついて に 義肢 装具 材料 の 非 金属 製 の プラスチック 以外 。
Baseline + Preordering	The non ‐ metallic materials of a prosthesis apparatus is made of plastics.
Organizers'	The non ‐ made of metal except for the plastic prosthesis apparatus material are discussed.
Reference	Artificial limb apparatus nonmental material other than plastic is discussed.

Example 2: Positive Impact of Preordering	
human judgment	1 1 1
Source	標記の新しいインライン流量計を開発した。
Preordered Source	開発 した を インライン 流量 計 新しい の 標記 。
Baseline + Preordering	We have developed a new in-line flowmeter.
Organizers'	The titled new in-line flowmeter was developed.
Reference	The titled new in‐line flowmeter was developed.

Table 5: Positive and negative effects of preordering

ample 2 shows a very natural structure, following the word order of the preordered Japanese source almost exactly.

5 Conclusion and Future Work

In this work we investigated several modifications and insertions to the pipeline of our baseline statistical machine translation system.

In the preprocessing phase we preordered the Japanese input text to compensate for order differences in the distant Japanese-English language pair. In the tuning phase of our decoder we evaluated different ways to include phrase similarity scores computed utilizing a neural network based compositional distributed semantics model. In the postprocessing phase we reranked the translation candidates for each sentence according to scores calculated with a recurrent neural network language model.

To our knowledge this is the first attempt to utilize neural network features directly integrated into the tuning and decoding process of a SMT system in a Japanese-English translation task. We therefore regard this work a first step towards improving Japanese to English translation systems by tightly coupling them with neural networks.

Our experiments show that preordering fails to yield the same improvements in the scientific paper domain that it has previously achieved on text from the legal/patent domain. We attribute this to the difference in the rigidness of possible formulations in these domains. The heuristic rules our preordering method is based on can better capture the more rigid sentences in the patent domain while they are prone to mismatches on the less rigid sentences in the scientific paper domain.

We conclude that neural networks utilized in machine translation should take word order into account and should be trained towards objectives that do not neglect syntactic features. The semantic similarity that can be captured by the model described in (Hermann and Blunsom, 2014) appears to be already sufficiently covered by the default features of our baseline system for the case of machine translation.

Employing the recursive neural network language model introduced by Mikolov et al. (Mikolov et al., 2011a) has proven successful for reranking translation candidates and significantly improved the RIBES score in our experiments.

Recent work (Devlin et al., 2014; Cho et al., 2014; Zhang et al., 2014; Liu et al., 2014) proves, there is a lot of potential in utilizing neural network based models in the machine translation pipeline. With the lessons learned from our work we hope for successful applications of this combination to the Japanese-English language pair in the future.

References

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011.

Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Katsuhiko Hayashi, Katsuhito Sudoh, Hajime Tsukada, Jun Suzuki, and Masaaki Nagata. 2013. Shift-reduce word reordering for machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Karl Moritz Hermann and Phil Blunsom. 2014. Multilingual models for compositional distributed semantics. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Sho Hoshino, Yusuke Miyao, Katsuhito Sudoh, and Masaaki Nagata. 2014. A preordering method robust to parsing errors for japanese-to-english statistical machine translation. In *Proceedings of the 20th Annual Meeting of the Association for Natural Language Processing*. (in Japanese).

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010a. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2010b. Head finalization: A simple reordering rule for sov languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*.

Hideki Isozaki, Natsume Kouchi, and Tsutomu Hirao. 2014. Dependency-based automatic enumeration of semantically equivalent word orders for evaluating japanese translations. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*.

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 the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*.

Taku Kudo, Hiroshi Ichikawa, and Hideto Kazawa. 2014. A joint inference of deep case analysis and zero subject generation for japanese-to-english statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *Journal of Machine Learning Research*, 32:1188–1196.

Shujie Liu, Nan Yang, Mu Li, and Ming Zhou. 2014. A recursive recurrent neural network for statistical machine translation. In *Proceedings of the 52th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics*.

Tomas Mikolov, Anoop Deoras, Stefan Kombrink, Lukàs Burget, and Jan Cernockỳ. 2011a. Empirical evaluation and combination of advanced language modeling techniques. In *Proceedings of the 12th Annual Conference of the International Speech Communication Association*, pages 605–608.

Tomas Mikolov, Stefan Kombrink, Anoop Deoras, Lukas Burget, and Jan Cernocky. 2011b. RNNLM – recurrent neural network language modeling toolkit. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop 2011 Demo Session*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Toshiaki Nakazawa, Hideya Mino, Isao Goto, Sadao Kurohashi, and Eiichiro Sumita. 2014. Overview of the 1st workshop on asian translation. In *Proceedings of the 1st Workshop on Asian Translation (WAT2014)*.

Richard Socher, Christopher D Manning, and Andrew Y Ng. 2010. Learning continuous phrase representations and syntactic parsing with recursive neural networks. In *Proceedings of the Advances in Neural Information Processing Systems 23 Deep Learning and Unsupervised Feature Learning Workshop*, pages 1–9.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4.

Kazushi Yoshida, Tomohiro Ohno, Yoshihide Kato, and Shigeki Matsubara. 2014. Japanese word reordering integrated with dependency parsing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*.

Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. 2014. Bilingually-constrained phrase embeddings for machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.