Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 3683–3691
Marseille, 11–16 May 2020

JASS: Japanese-specific Sequence to Sequence Pre-training for Neural Machine Translation

Zhuoyuan Mao[†], Fabien Cromieres[†⋆], Raj Dabre[‡⋆], Haiyue Song[†], Sadao Kurohashi[†]
[†]Kyoto University
[‡]National Institute of Information and Communications Technology
{zhuoyuanmao, fabien, song, kuro}@nlp.ist.i.kyoto-u.ac.jp, raj.dabre@nict.go.jp

Abstract

Neural machine translation (NMT) needs large parallel corpora for state-of-the-art translation quality. Low-resource NMT is typically addressed by transfer learning which leverages large monolingual or parallel corpora for pre-training. Monolingual pre-training approaches such as MASS (MAsked Sequence to Sequence) are extremely effective in boosting NMT quality for languages with small parallel corpora. However, they do not account for linguistic information obtained using syntactic analyzers which is known to be invaluable for several Natural Language Processing (NLP) tasks. To this end, we propose JASS, Japanese-specific Sequence to Sequence, as a novel pre-training alternative to MASS for NMT involving Japanese as the source or target language. JASS is joint BMASS (Bunsetsu MASS) and BRSS (Bunsetsu Reordering Sequence to Sequence) pre-training which focuses on Japanese linguistic units called bunsetsus. In our experiments on ASPEC Japanese–English and News Commentary Japanese–Russian translation we show that JASS can give results that are competitive with if not better than those given by MASS. Furthermore, we show for the first time that joint MASS and JASS pre-training gives results that significantly surpass the individual methods indicating their complementary nature. We will release our code, pre-trained models and bunsetsu annotated data as resources for researchers to use in their own NLP tasks.

Keywords: pre-training, neural machine translation, bunsetsu, low resource

1. Introduction

Encoder-decoder based neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015), and in particular, the Transformer model (Vaswani et al., 2017) have led to a large jump in the quality of automatic translation over previous approaches such as Statistical Machine Translation (Koehn, 2004). One of the drawbacks of NMT is that it requires large parallel corpora for training robust and high quality translation models. This strongly limits its usefulness for many language pairs and domains for which no such large corpora exist.

The most popular way to solve this issue is to leverage monolingual corpora, which are much easier to obtain (as compared to parallel corpora) for most languages and domains. This can be done either by backtranslation (Sennrich et al., 2016a; Hoang et al., 2018; Edunov et al., 2018) or by pre-training. Pre-training consists in initializing some or all of the parameters of the model through tasks that only require monolingual data. One can pre-train the word embeddings of the model (Qi et al., 2018) or the encoder and decoders (Zoph et al., 2016). Pre-training has recently become the focus of much research after the success of methods such as BERT (Devlin et al., 2019), ELMO (Peters et al., 2018) or GPT (Radford, 2018) in many NLP tasks. However, these methods were not designed to be used for NMT models in the sense that BERT-like models are essentially language models and not sequence to sequence models. (Song et al., 2019) have obtained new state-of-the-art results for NMT in low-resource settings by addressing these issues and providing a pre-training method for sequence to sequence models: MASS (MAsked Sequence to Sequence).

Another way to overcome the scarcity of parallel data is to provide the model with more "linguistic knowledge", such as language-specific information. Works such as (Sennrich and Haddow, 2016; Murthy et al., 2019; Zhou et al., 2019) have shown that such information could improve results. However, because NMT models are end-to-end sequence to sequence models, the manner in which such linguistic hints should be provided is not always clear.

In this paper, we argue that pre-training provides an ideal framework both for leveraging monolingual data and improving NMT models with linguistic information. Our setting focuses on the translation between language pairs involving Japanese. Japanese is a language for which very high quality syntactic analyzers have been developed (Kurohashi et al., 1994; Morita et al., 2015). On the other hand, large parallel corpora involving Japanese exist only for a few language pairs and domains. As such it is critical to leverage both monolingual data and the syntactic analyses of Japanese for optimal translation quality.

Our pre-training approach is inspired by MASS, but with more linguistically motivated tasks. In particular, we add syntactic constraints to the sentence-masking process of MASS and dub the resulting task BMASS[1]. We also add a linguistically-motivated reordering task that we dub BRSS (Bunsetsu Reordering Sequence to Sequence). We combine these two tasks to obtain a novel pre-training method tailored for Japanese that we call JASS (Japanese-specific Sequence to Sequence).

We experiment on the ASPEC Japanese–English dataset in a variety of settings ranging from 1000 to 1,000,000 parallel sentences. We also experiment with a realistic setting for a difficult language pair, namely, Japanese-Russian. Our results show that JASS by itself is already at least as good

⋆ represents equal contribution to this work

[1] For Bunsetsu-MASS, bunsetsus are one of the elementary syntactic components of Japanese

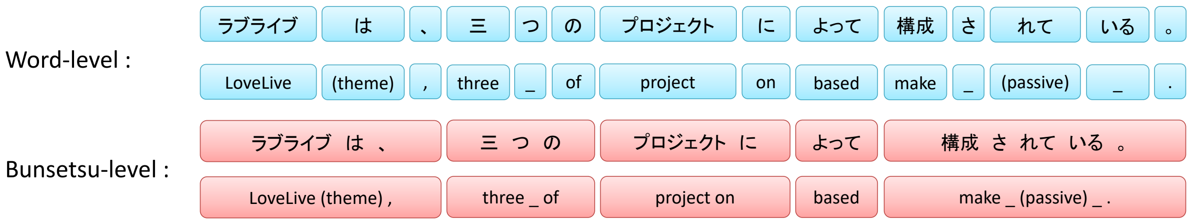

Figure 1: Word and bunsetsu segmentations for a Japanese sentence with the meaning “LoveLive is made of three projects.” Below is the English version translated word-for-word, where “_” represents the meaningless segmented parts.

as and often better than using the state-of-the-art MASS pre-training. Furthermore, we show that combining MASS and JASS lead to further improvements of up to +1.7 BLEU in low resource settings.

To the best of our knowledge, this is the first time that syntactic information is used in such a pre-training setting for NMT. We make our code and pre-trained models publicly available[2].

The contributions of this paper are as follows:

- JASS: a novel linguistically motivated pre-training method for NMT involving Japanese.
- Showing how MASS and JASS complement each other indicating that combining multiple types of pre-training techniques can yield better results than using only one type of pre-training.
- An empirical comparison of MASS and JASS for ASPEC Japanese–English translation in several data size settings to identify situations where each technique can be most useful.
- Verifying that pre-training is a good way to feed linguistic information into to a model.
- Pre-trained models, code and annotated data as resources for reproducibility and public use.

2. Related Work

Pre-training based approaches are essentially transfer learning approaches where we leverage an external source of data to train a model whose components can be used for NLP tasks which do not have abundant data. In the context of NMT, cross-lingual transfer (Zoph et al., 2016) was shown to be most effective to improve Hausa-English translation when a pre-trained French-English NMT model was fine-tuned on Hausa-English data. While this work focused on strongly pre-training the English side decoder, (Dabre et al., 2019) showed that pre-training the encoder is also useful through experiments on fine-tuning an English–Chinese model on a small multi-parallel English–X (7 Asian languages) data. All these works rely on bilingual corpora but our focus is on leveraging monolingual corpora that are orders of magnitude larger than bilingual corpora.

Pre-trained models such as BERT (Devlin et al., 2019), ELMO (Peters et al., 2018), XLNET (Yang et al., 2019) and GPT (Radford, 2018) have proved very useful for tasks such as Text Understanding, but have a limited application to NMT, as they only pre-train the encoder side of a transformer. Pre-training schemes more suitable to NMT have been proposed by (Lample and Conneau, 2019), (Ren et al., 2019) and (Song et al., 2019). In particular, (Song et al., 2019) obtained state-of-the-art results with their “MASS” pre-training scheme. MASS allows for the simultaneous pre-training of the encoder and decoder and hence is the most useful for NMT. However, MASS does not consider the linguistic properties of language when pre-training whereas our objective is to show that linguistically motivated pre-training can be complementary to MASS. Our research is motivated by previous research (Kawahara et al., 2017) for Japanese NLP which showed that linguistic annotations from the syntactic analyzers such as Juman (Morita et al., 2015) and KNP (Kurohashi et al., 1994) are extremely important.

Pre-ordering consists of pre-processing a sentence so that its word-order is more similar to that of its expected translation. It has been a popular technique for Statistical Machine Translation since the early work of (Collins et al., 2005). Although initial research (Du and Way, 2017) had concluded that pre-ordering had limited usefulness for NMT, it has been shown more recently that it can improve translation quality, especially in the case of low-resource languages. (Murthy et al., 2019) showed that pre-ordering English to Indic language word order is beneficial when performing transfer learning via fine-tuning. (Zhou et al., 2019) showed that leveraging structural knowledge for creating the psuedo Japanese-ordered English by pre-ordering English from SVO to SOV improves Japanese–English translation. Our work will try to incorporate similar ideas directly in the pre-training process. On the related matter of the usefulness of linguistic information for NMT, (Sennrich and Haddow, 2016) also showed how linguistic annotations can help improve German–English translation.

3. Background: MASS and Bunsetsu

Central to our work are Bunsetsu and MASS which we explain as below.

3.1. Bunsetsu

Bunsetsus are syntactic components of Japanese sentences. They are roughly equivalent to the noun chunks or verb chunks in English syntax. They constitute a minimal unit of meaning. Japanese segmenters can segment a Japenese sentence in words or in bunsetsus, but the concept of “word”

[2] https://github.com/Mao-KU/JASS

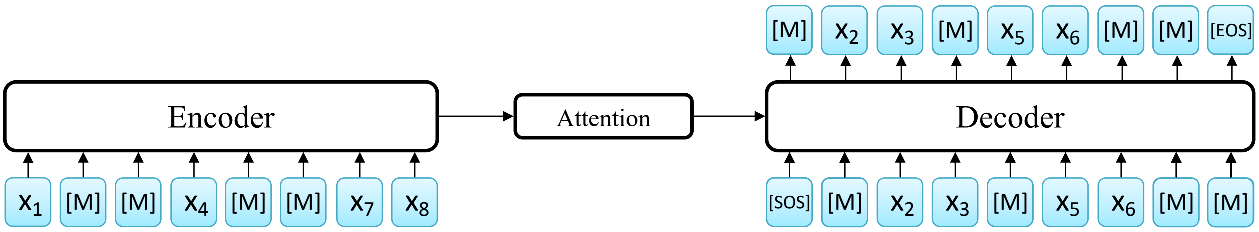
Encoder
Attention
Decoder
X1 [M] [M] X4 [M] [M] X7 X8
[SOS] [M] X2 X3 [M] X5 X6 [M] [M]
[M] X2 X3 [M] X5 X6 [M] [M] [EOS]

Figure 2: Sequence to Sequence Structure for MASS, where x_i represents a token and x_2, x_3 and x_5, x_6 are consecutive tokens to be masked/predicted. In the case of BMASS pre-training, x_2, x_3 and x_5, x_6 are bunsetsus.

is ambiguous for writing systems that do not use word-separators (like spaces), like Japanese. Bunsetsus are also more likely to correspond to a well-defined entity or concept than words. The conceptual difference between using word level and bunsetsu level segmentation is shown in Figure 1 for the Japanese sentence with the meaning "Love-Live is made of three projects." Note that each bunsetsu contains some self contained information and some case marker which can indicate its relation with another bunsetsu.

3.2. MASS

MASS is a pre-training method for NMT proposed by (Song et al., 2019). In MASS pre-training the input is a sequence of tokens where a part of the sequence is masked and the output is a sequence where the masking is inverted. Consider $x \in \mathcal{X}$ which is a sequence of tokens where $\mathcal{X}$ is a monolingual corpus. Consider $C = [[p_1, p_2], [p_3, p_4], ...[p_n, p_{n+1}]]$ where $0 < p_1 \leq p_2 \leq p_3 \leq p_4 \leq ...p_n \leq p_{n+1} \leq len(x)$ and $len(x)$ is the number of tokens in sentence x. We denote by x^C the masked sequence where tokens in positions from p_1 to p_2, p_3 to p_4 and so on until p_n to p_{n+1} in x are replaced by a special token $[M]$. $x^{!C}$ is the invert masked sequence where tokens in positions other than the aforementioned fragments are replaced by the mask token $[M]$. MASS is a pre-training objective that predicts the masked fragments in x using an encoder-decoder model where x^C is the input to the encoder and $x^{!C}$ is the reference for the decoder. The log likelihood objective function is:

$$\begin{aligned} \mathcal{L}_{mass}(\mathcal{X}) &= \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} \log P\left(x^{!C}|x^C;\theta\right) \qquad (1) \\ &= \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} \log \prod_{t\in C} P\left(x_t^{!C}|x_{<t}^{!C}, x^C;\theta\right) \end{aligned}$$

where $x^C_{<t}$ indicates the preceding tokens before t in x^C and θ is set of model parameters. The hyper-parameter for MASS is the number of tokens to be masked. Refer to Figure 3-b for a training pair example for MASS.

4. Proposed Method: JASS

JASS (Japanese-specific Sequence to Sequence pre-training) is an extension of the original MASS method to incorporate linguistic information in addition to reordering based pre-training (Zhang and Zong, 2016). It is a combination of two sub-methods, BMASS (Bunsetsu-based MAsked Sequence to Sequence pre-training) and BRSS (Bunsetsu Reordering Sequence to Sequence pre-training).

4.1. BMASS

In MASS, a NMT model is trained by making it predict random parts of a sentence given their context. Instead of random parts we are interested in making the model predict a set of bunsetsus given the contextual bunsetsus. We expect this will let the model learn about the important concept of bunsetsu, as well as focus its training on predicting meaningful subsequences instead of random ones.

More precisely, we propose BMASS (Bunsetsu-based MAsked Sequence to Sequence pre-training), which leverages syntactic parses of Japanese monolingual data for sequence to sequence pre-training. To perform BMASS, we modify the mask C in Equation 1 where the position spans p_1 to p_2, p_3 to p_4 and so on until p_n to p_{n+1} indicate the start and end of bunsetsus in a Japanese sentence. Consequently we denote the BMASS loss as $\mathcal{L}_{bmass}$. The main difference between MASS and BMASS is that in MASS we mask random token spans whereas in BMASS we mask tokens spans that only cover bunsetsus. The number of bunsetsus to be masked constitutes a hyper-parameter for BMASS.

Refer to Figure 3-c for a training pair example for BMASS, which may be contrasted with the MASS example in figure 3-b.

4.2. BRSS

BRSS (Bunsetsu Reordering based Sequence to Sequence) roughly consist in training the NMT system with re-ordered Japanese. We expect that this will let the system learn the structure of Japanese language, as well as prepare it for the reordering operation it will have to perform when translating to a language with different grammar.

4.2.1. Bunsetsu-based Reordering

We first define here a simple process for re-ordering a (typically SOV) Japanese sentence into a "SVO-ordered Japanese" pseudo-sentence. We will then use this reordered sentence in section 4.2.2. for our BRSS pre-training.

There exist several previous works about reordering a SOV-ordered sentence to a SVO-ordered sentence (Katz-Brown and Collins, 2008; Hoshino et al., 2014). In our case,

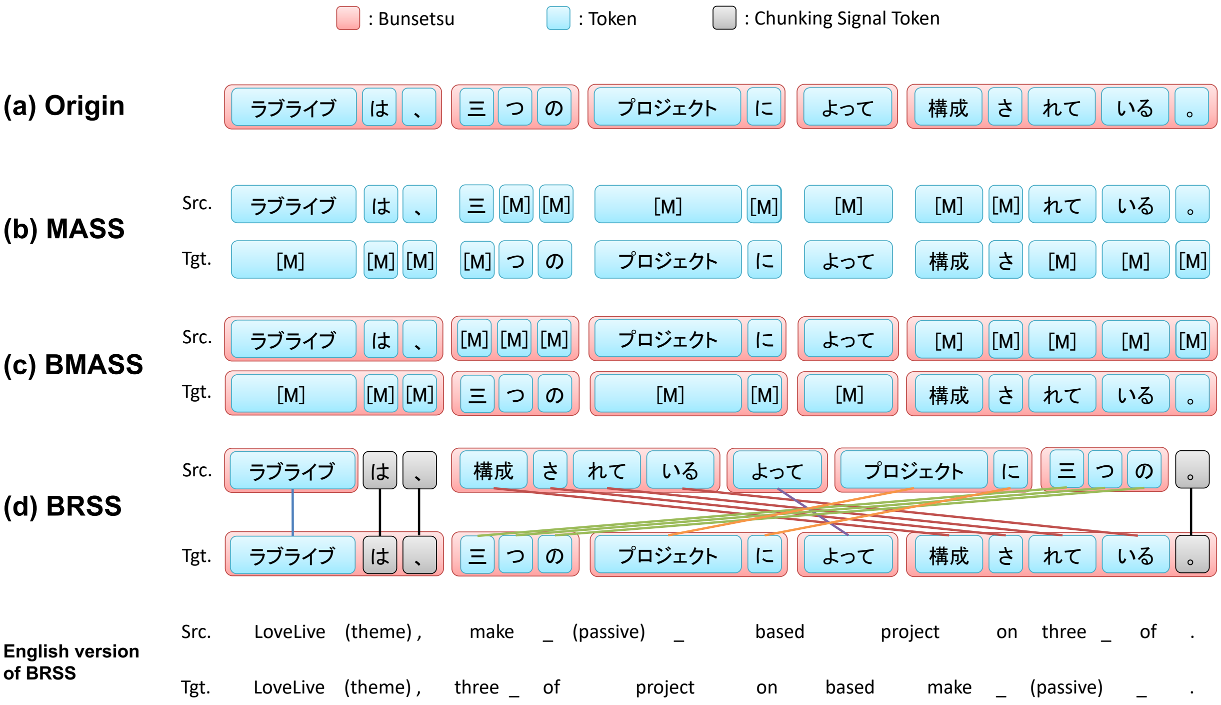
: Bunsetsu
: Token
: Chunking Signal Token
(a) Origin
ラブライブ は 、 三 つ の プロジェクト に よって 構成 さ れて いる 。
(b) MASS
Src.
ラブライブ は 、 三 [M] [M] [M] [M] [M] [M] [M] れて いる 。
Tgt.
[M] [M] [M] [M] つ の プロジェクト に よって 構成 さ [M] [M] [M]
(c) BMASS
Src.
ラブライブ は 、 [M] [M] [M] プロジェクト に よって [M] [M] [M] [M] [M]
Tgt.
[M] [M] [M] 三 つ の [M] [M] [M] 構成 さ れて いる 。
(d) BRSS
Src.
ラブライブ は 、 構成 さ れて いる よって プロジェクト に 三 つ の 。
Tgt.
ラブライブ は 、 三 つ の プロジェクト に よって 構成 さ れて いる 。
English version of BRSS
Src.
LoveLive (theme) , make _ (passive) _ based project on three _ of .
Tgt.
LoveLive (theme) , three _ of project on based make _ (passive) _ .

Figure 3: An example of source and target for MASS, BMASS, BRSS with the meaning “LoveLive is made of three projects.”

in order to leverage bunsetsu units in Japanese consistently with BMASS, we propose Bunsetsu-based Reordering, which is able to generate a SVO-ordered Japanese sentence while retaining syntactic information at the bunsetsu-level. Bunsetsu-based Reordering is implemented by the following steps:

- Split the Japanese into several chunks by chunking signal tokens. Specifically, chunking signal tokens includes the punctuation and ‘は’ which mentions the theme in a Japanese sentence
- Reverse the order of the bunsetsus in each chunk

4.2.2. Bunsetsu Reordering Sequence to Sequence Pre-training

We build up our BRSS (Bunsetsu Reordering Sequence to Sequence) pre-training on the basis of Bunsetsu-based Reordering as mentioned above. Refer to Figure 3-d for a training pair example for BRSS. The pre-training objective here is a deshuffling or un-reordering task which reconstructs the original sentence from the reordered sentence.

The bunsetsu reordering process described above in section 4.2.1. allows us to produce an artificial “SVO-Japanese” sentence for each sentence in a training monolingual corpora. We then have two choices for the pre-training procedure. We can make the NMT system predict the artificial reordered SVO sentence given the original. Alternatively, we can make it predict the original given the reordered one. We experiment with both options in the following section. These 2 pre-training directions are denoted as BRSS.F (reordered to original) and BRSS.R (original to reordered). Following the notation in MASS, we define the log likelihood objective function of BRSS as follows:

$$\mathcal{L}_{brss}(\mathcal{X}) = \mathcal{L}_{brss.f}(\mathcal{X}) \ or \ \mathcal{L}_{brss.r}(\mathcal{X}) \tag{2}$$

$$\mathcal{L}_{brss.f}(\mathcal{X}) = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} \log P\left(x|x^{reordered};\theta\right) \tag{3}$$

$$\mathcal{L}_{brss.r}(\mathcal{X}) = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} \log P\left(x^{reordered}|x;\theta\right) \tag{4}$$

Note that the equivalent reordering based pre-training mechanism is one where we randomly shuffle a sentence at the word level. However, this does not focus on learning any kind of specific linguistic reordering and so we do not explore it in this work.

4.3. JASS

In the previous sections, we have defined two pre-training procedures: BMASS and BRSS. Our actual pre-training will consist in a joint execution of these two pre-training. We call the resulting pre-training JASS (JApanese-specific Sequence to Sequence) pre-training. The pre-training objective for JASS is therefore:

$$\mathcal{L}_{jass}(\mathcal{X}) = \mathcal{L}_{bmass}(\mathcal{X}) + \mathcal{L}_{brss}(\mathcal{X}) \tag{5}$$

where $\mathcal{X}$ represents the monolingual corpus of Japanese. We expect BMASS to learn syntactic knowledge and BRSS

	Language	Dataset	Size
Mono	Ja	Common Crawl	22M
	En	News Crawl	22M
	Ru	News Crawl	22M
Parallel	Ja-En	ASPEC-JE	3M
	Ja-Ru	JaRuNC	10K

Table 1: Overview of data

to learn word ordering knowledge.

Because JASS has been specifically designed for Japanese, and we have not yet considered equivalents for other languages, we also mix JASS pre-training for Japanese with MASS pre-training for the other language involved in the translation.

In practice, we therefore designate by JASS the pre-training of the NMT system that uses Japanese monolingual data with BMASS and BRSS objectives, and "other language" monolingual data with MASS objective.

We can also consider using Japanese monolingual data with a combination of BMASS, BRSS and MASS objectives, which we dub MASS+JASS in the following sections.

5. Experimental Settings

In this section, we evaluate our pre-training methods on 4 translation directions: Japanese-to-English (Ja-En), English-to-Japanese (En-Ja), Japanese-to-Russian (Ja-Ru) and Russian-to-Japanese (Ru-Ja). Specifically, we monitor the performance of our pre-training methods on both simulated low-resource and high-resource scenarios involving ASPEC Japanese–English translation (Nakazawa et al., 2015). We also test our methods on a realistic low-resource scenario involving News Commentary Japanese–Russian translation[3] (Imankulova et al., 2019).

5.1. Datasets and Pre-processing

We use both the monolingual data and parallel data for pre-training and the parallel data for fine-tuning. Refer to Table 1 for an overview.

5.1.1. Parallel Data

We use scientific abstracts domain ASPEC parallel corpus (Nakazawa et al., 2016) for Japanese–English translation and the news commentary domain JaRuNC parallel corpus (Imankulova et al., 2019) for Japanese–Russian translation.

5.1.2. Monolingual data

We use monolingual data containing 22M Japanese, 22M English and 22M Russian sentences randomly sub-sampled from Common Crawl dataset and News crawl[4] dataset from the official WMT monolingual training data[5] for pre-training. Each side of the parallel data used in fine-tuning is also incorporated into the monolingual data for pre-training. Specifically, for Japanese and English, 3M sentences from each side of the parallel data is added to the monolingual data while for Japanese and Russian, 10K sentences from each side of the parallel data is also used in pre-training. This results in 50M monolingual sentences for Japanese and English, and 45M monolingual sentences for Japanese and Russian. Given that our pre-training objective works at the monolingual level and that the three languages have different scripts and thus have few common words, we believe this to be a fair pre-training data setting.

5.1.3. Pre-processing

We tokenize the monolingual data by using the Moses tokenizer[6] for En and Ru, and the Jumanpp tokenizer[7] for Ja. We get the bunsetsu information by using KNP[8]. Sentences with length over 175 tokens are removed. For each language pair, we built a joint vocabulary with 60,000 subword units via Byte-Pair Encoding(Sennrich et al., 2016b). Considering the discrepancy of the domain between pre-training dataset and fine-tuning dataset, we oversample the fine-tuning dataset when learning BPE codes. Since some English alphabets appear in the Japanese and Russian corpora, the BPE codes are learned jointly from the concatenation of the corpora for each language pair. As we do multi-task pre-training, each sentence is prepended with a task token $[MASS]$, $[BMASS]$ or $[RSS]$ and a language token $[Ja]$, $[En]$, or $[Ru]$. This ensures that the model learns to distinguish between different pre-training objectives and languages.

5.2. Model Training and Evaluation Settings

For the NMT model, we experiment with a Transformer (Vaswani et al., 2017) having 6 layers for both the encoder and the decoder. We implement our approaches on top of the OpenNMT[9] transformer implementation.

Model	1K	10K	20K	100K	1M
Transformer-big	0.40	2.56	9.53	22.72	29.50
Transformer-base	0.33	1.79	8.21	21.34	29.06

Table 2: BLEU on Ja-En (ASPEC-JE)

OpenNMT provides two default hyperparameters settings that differ in the size of layer used and the number of attention heads, namely, "base" and "big". Although we could have expected the smaller model to be a better fit for low-resource training, we found out the opposite. Table 2 contains our preliminary experiments where the Transformer

[3] Neither Japanese nor Russian are low-resource languages, but Ja-Ru can be regarded as a low-resource language pair because of the limited amount of the parallel data.

[4] The pre-training will be very effective if the domains of the pre-training and fine-tuning dataset are similar(Raffel et al., 2019). However, in order to obtain a general pre-trained model for NMT, we choose the monolingual data from Common Crawl and News Crawl.

[5] http://www.statmt.org/wmt19/translation-task.html

[6] https://github.com/moses-smt/mosesdecoder

[7] https://github.com/ku-nlp/jumanpp

[8] https://github.com/ku-nlp/pyknp

[9] https://github.com/OpenNMT/OpenNMT-py

in the big setting outperforms Transformer in the base setting for both high-resource and low-resource scenarios for Japanese–English translation.
Therefore, we implement our pre-training methods and fine-tuning using the Transformer-big setting, which consists of a 6-layer encoder and a 6-layer decoder, with the length of 1024 for hidden size, the length of 4096 for feed-forward size, dropout rate of 0.3 and attention heads of 16. A learning-rate of 10^{-4} is used both for pre-training and fine-tuning, and all the pre-training tasks are implemented on 8 TITAN X (Pascal) GPU cards until convergence with a batch-size of 2048 for each GPU while single GPU is used for fine-tuning. The checkpoint with the highest accuracy is selected for fine-tuning. We use BLEU (Papineni et al., 2002) to implement the evaluation. We do early stopping if no improvement on development-set within 5 checkpoints, and the checkpoint with the best BLEU performance on development-set is selected for evaluation.
For multi-task pre-training, data is randomly shuffled so that even in each mini-batch, different pre-training objectives will appear, corresponding to a real joint pre-training. We evaluate the statistical significance of our BLEU scores by bootstrap resampling (Koehn, 2004).

5.3. Pre-trained models

We pre-train our NMT models by leveraging the monolingual data of the source and target languages. For Japanese we use MASS as well as JASS, while for English and Russian, we only use MASS as the pre-training objective. In particular we pre-train the following models:

- **MASS:** We use the same settings as in (Song et al., 2019) for pre-training.
- **BMASS:** Similar to MASS we mask half the bunsetsus in a sentence during pre-training.
- **BRSS:** Using our approach in Section 4.2. we pre-train on SVO–SOV (BRSS.F) Japanese sentence pairs.
- **JASS:** Multi-task training of BMASS and BRSS.
- **MASS+BMASS:** Multi-task training of MASS and BMASS.
- **MASS+BRSS:** Multi-task training of MASS and BRSS.
- **JASS+MASS:** Multi-task training of BMASS, BRSS and MASS.

5.4. Fine-tuning on NMT

As mentioned above, we validate the effectiveness of our pre-training methods by 4 fine-tuning tasks, which are Ja-En, En-Ja, Ja-Ru, Ru-Ja. We train the following models by fine-tuning the pre-trained models:

- **ASPEC Ja–En and En–Ja:** Japanese to English and English to Japanese models using from 1K to 1M[10] parallel sentences.
- **NC Ja–Ru and Ru–Ja:** Japanese to Russian and Russian to Japanese models using available 12,356 training pairs .

We compare these models with baselines which do not use pre-training.

6. Results & Analysis

We now give the results for Japanese–English and Japanese–Russian translation. All the results are reported on the official test sets provided by the 2019 edition of the Workshop on Asian Translation(WAT)[11].

6.1. Pre-training Accuracy

Setting	En+Ja	Ru+Ja
MASS	71.18	72.35
BMASS	73.76	73.98
BRSS.F	84.82	84.89
JASS(BMASS+BRSS.F)	81.53	81.63
MASS+BMASS	72.33	-
MASS+BRSS.F	79.56	-
MASS+JASS	78.62	78.85

Table 3: Pre-training accuracy, which is the 1-gram accuracy of the pre-trained model

Pre-training accuracy can be an indicator of the learning difficulty. The pre-training objectives should not be too easy or too difficult. As shown in Table 3, for Japanese–English pre-training, BRSS is the easiest for the neural network while MASS is of the highest difficulty. Moreover, it can be found that the accuracy of a pre-training objective does not vary a lot from one language pair to another. As easy and difficult are subjective we use pre-training accuracy as one of the indicators of the difficulty and hence the usefulness of our pre-training approach. MASS+JASS gives the best BLEU performance in most of our experiments and thus we hypothesize that there is no perfect pre-training method and thus one should explore a variety of methods for a given language pair.

6.2. Fine-tuning Results

Tables 4, 5, 6 contain the results of fine-tuning the pre-trained models for Japanese–English and English–Japanese translation. Our pre-training methods, BMASS and BRSS, clearly improved on the strictly-supervised baselines and fine-tuning gives results comparable to those of MASS, which validate the effectiveness of our Japanese-specific objectives for pre-training. In Table 4, 5, we observe that JASS significantly outperforms ($p < 0.05$) MASS, when parallel corpora sizes from 3K to 50K are used. In other size settings, JASS is competitive with if not significantly better than MASS and this demonstrates that linguistically motivated pre-training can be an alternative to language-agnostic pre-training. In Table 4, 5, 6, the joint pre-training

[10] We limit ourselves to 1M sentences because the remaining 2M sentences are relatively noisy and most of previous research mainly relies on the best 1M sentences for best translation quality.

[11] http://lotus.kuee.kyoto-u.ac.jp/WAT/WAT2019/index.html

Model	1K	3K	6K	10K	20K	50K	100K	200K	500K	1M
Supervised(Transformer-big)	0.40	1.30	1.55	2.56	9.53	17.56	22.72	25.51	27.92	29.50
MASS	5.34	9.89	12.28	15.16	18.65	22.28	24.86	26.67	28.85	29.63
BMASS	4.06	8.49	11.70	14.32	18.56	22.30	24.65	26.77	28.55	29.72
BRSS.F	3.29	8.61	12.12	14.75	18.40	22.07	24.55	26.53	28.71	29.53
BRSS.R	3.00	7.36	11.02	13.74	17.30	21.80	24.52	26.56	28.45	29.52
JASS(BMASS+BRSS.F)	5.18	10.06	13.49†	15.55†	19.12†	22.85†	**25.20**	26.88	28.62	29.67
MASS+BMASS	4.64	8.74	12.39	14.22	18.18	22.21	24.86	26.68	28.96	29.80
MASS+BRSS.F	5.88†	**10.78**†	13.53†	15.99†	19.01	22.67	24.90	26.75	**28.98**	**29.88**
MASS+JASS	**6.28**†	10.72†	**13.97**†	**16.09**†	**19.34**†	**23.15**†	24.99	**27.09**†	28.82	29.49

Table 4: BLEU scores for simulated low/high-resource settings for Ja-En ASPEC translation using 3K to 1M parallel sentences for fine-tuning. Results better than MASS with statistical significance $p < 0.05$ are marked with †

Model	1K	3K	6K	10K	20K	50K	100K	200K	500K	1M
Supervised(Transformer-big)	0.75	1.49	2.21	3.68	11.52	20.95	27.94	32.71	38.89	40.26
MASS	5.81	11.02	15.29	18.11	21.57	27.91	31.62	34.88	38.97	**41.16**
BMASS	5.03	9.77	13.40	17.25	21.14	27.10	30.97	34.90	39.00	40.50
BRSS.F	3.54	10.30	14.86	17.67	21.64	27.48	31.22	34.88	38.21	40.43
BRSS.R	4.31	9.77	14.25	16.89	20.81	26.34	30.69	33.91	38.49	40.27
JASS(BMASS+BRSS.F)	5.54	11.37	15.91†	18.50†	22.18†	27.27	31.05	34.72	38.89	40.64
MASS+BMASS	5.20	10.00	14.37	17.44	21.53	27.24	30.98	**35.14**	**39.40**†	40.65
MASS+BRSS.F	6.53†	12.04†	15.79†	18.95†	22.32†	27.32	**31.63**	34.69	38.85	41.09
MASS+JASS	**6.82**†	**12.57**†	**16.22**†	**19.20**†	**23.00**†	**28.09**	31.43	34.81	38.43	40.79

Table 5: BLEU scores for simulated low/high-resource settings for Ja-En ASPEC translation using 3K to 1M parallel sentences for fine-tuning. Results better than MASS with statistical significance $p < 0.05$ are marked with †

Model	Ja-Ru	Ru-Ja
Supervised(Transformer-big)	0.50	0.72
MASS	0.96	2.84
BMASS	0.97	2.77
BRSS.F	0.85	2.36
JASS(BMASS+BRSS.F)	**1.20**	3.08
MASS+JASS	1.07	**3.45**†

Table 6: BLEU scores for Ja-Ru translation on JaRuNC. Results better than MASS with statistical significance $p < 0.05$ are marked with †

of MASS and JASS (BMASS+BRSS) leads to the highest BLEU scores ($p < 0.05$) on most settings, which indicates that JASS is not just a alternative to MASS, but could be complementary to MASS. Human evaluation of the translations from both systems should shed more light on this. We leave this for future work.

Finally, it can be seen that pre-training outperforms the supervised baselines in almost all data scenarios which shows that for neural machine translation, pre-training is a valuable strategy especially for low-resource scenarios. Moreover, since pre-training enables the encoder-decoder to learn an implicit language model it can help overcome the scarcity of language modeling information in parallel corpora. Given the success of JASS in low-resource scenarios, we believe that it is absolutely necessary to leverage language specific information during pre-training.

Unfortunately, in Table 6, the improvement contributed by bilingual pre-training is limited on JaRuNC. Japanese–Russian is a difficult language pair, the fine-tuning data is small and the news commentary domain is much harder than the ASPEC domain. As such we feel that multi-lingual pre-training and fine-tuning mechanisms might help alleviate this issue as shown by (Imankulova et al., 2019). This too, we leave for future work.

6.2.1. BRSS.F or BRSS.R?

Although we mentioned 2 pre-training methods involving reordering which are named as BRSS.F and BRSS.R, we mostly experimented with joint pre-training using BRSS.F. In order to demonstrate this choice, we give the fine-tuning results of BRSS.F and BRSS.R on ASPEC as shown in Table 4, 5. We observe that BRSS.F outperforms BRSS.R in most cases, regardless of translation direction, which is probably because BRSS.F is able to pre-train a better decoder to generate natural language, even though the reordering described by BRSS.R should be more appropriate for Ja-En translation. Currently, we do not have any detailed explanation why BRSS.F is consistently better than BRSS.R and we will investigate this in the future.

7. Conclusion

In this paper we proposed JASS (Japanese-specific sequence to sequence) pre-training which are novel pre-training alternatives to MASS for neural machine translation involving Japanese as the source or target language. Our work is aimed at leveraging abundant monolingual data and syntactic analyses provided by analyzers so that the pre-training stage becomes aware of language structure. Our experiments on ASPEC Japanese–English translation and News Commentary Japanese–Russian translation have shown that JASS, which leverages syntactic parsing knowledge from the KNP parser, outperform MASS, which is language agnostic, in many low-resource settings. Furthermore, we show that the combination of MASS and JASS yields significantly better results than the individual pre-training methods. This demonstrates the effectiveness of our methods and the necessity to inject language-specific information into the pre-training objective. We have publicly released our code and models. To the best of our knowledge, this is the first time that linguistic information has been used for pre-training a NMT system. Our positive results show that the pre-training step is an appropriate place to provide linguistic hints to a NMT system.

We are now working on several directions for improving and broadening our approach. Pre-training methods do not often consider domain differences within the data and so in the future, we will try to address domain adaptation in order to enhance the impact of fine-tuning on in-domain data. In particular we find the multi-stage training approach (Imankulova et al., 2019; Dabre et al., 2019) most relevant in this direction. We will also work on determining the impact of multi-task pre-training using a combination of a wide variety of pre-training approaches that focus on different aspects of language structure. We might also apply ideas similar as the ones developped here to different languages. We also note that (Raffel et al., 2019) has recently shown that many NLP tasks such as Text Understanding could be reformulated as Text-to-Text tasks. This broadens a lot the domain of usefulness of text-to-text pre-training tasks such as ours, and we will be interested in evaluating our approach on a wider range of NLP tasks.

8. References

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, USA, May.

Collins, M., Koehn, P., and Kučerová, I. (2005). Clause restructuring for statistical machine translation. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 531–540. Association for Computational Linguistics.

Dabre, R., Fujita, A., and Chu, C. (2019). Exploiting multilingualism through multistage fine-tuning for low-resource neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1410–1416, Hong Kong, China, November. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Du, J. and Way, A. (2017). Pre-reordering for neural machine translation: Helpful or harmful? *The Prague Bulletin of Mathematical Linguistics*, 108(1):171–182.

Edunov, S., Ott, M., Auli, M., and Grangier, D. (2018). Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium, October-November. Association for Computational Linguistics.

Hoang, V. C. D., Koehn, P., Haffari, G., and Cohn, T. (2018). Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia, July. Association for Computational Linguistics.

Hoshino, S., Soyer, H., Miyao, Y., and Aizawa, A. (2014). Japanese to English machine translation using preordering and compositional distributed semantics. In *Proceedings of the 1st Workshop on Asian Translation (WAT2014)*, pages 55–63, Tokyo, Japan, October. Workshop on Asian Translation.

Imankulova, A., Dabre, R., Fujita, A., and Imamura, K. (2019). Exploiting out-of-domain parallel data through multilingual transfer learning for low-resource neural machine translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 128–139, Dublin, Ireland, 19–23 August.

Katz-Brown, J. and Collins, M. (2008). Syntactic reordering in preprocessing for japanese → english translation: MIT system description for NTCIR-7 patent translation task. In *Proceedings of the 7th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access, NTCIR-7, National Center of Sciences, Tokyo, Japan, December 16-19, 2008*.

Kawahara, D., Hayashibe, Y., Morita, H., and Kurohashi, S. (2017). Automatically acquired lexical knowledge improves Japanese joint morphological and dependency analysis. In *Proceedings of the 15th International Conference on Parsing Technologies*, pages 1–10, Pisa, Italy, September. Association for Computational Linguistics.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

Kurohashi, S., Nakamura, T., Matsumoto, Y., and Nagao, M. (1994). Improvements of Japanese morphological analyzer JUMAN. In *Proceedings of the Inter-*

national Workshop on Sharable Natural Language Resources, pages 22–28.

Lample, G. and Conneau, A. (2019). Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Morita, H., Kawahara, D., and Kurohashi, S. (2015). Morphological analysis for unsegmented languages using recurrent neural network language model. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2297, Lisbon, Portugal, September. Association for Computational Linguistics.

Murthy, R., Kunchukuttan, A., and Bhattacharyya, P. (2019). Addressing word-order divergence in multilingual neural machine translation for extremely low resource languages. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3868–3873, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Nakazawa, T., Mino, H., Goto, I., Neubig, G., Kurohashi, S., and Sumita, E. (2015). Overview of the 2nd Workshop on Asian Translation. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 1–28, Kyoto, Japan, October.

Nakazawa, T., Yaguchi, M., Uchimoto, K., Utiyama, M., Sumita, E., Kurohashi, S., and Isahara, H. (2016). Aspec: Asian scientific paper excerpt corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, pages 2204–2208, Portorož, Slovenia, May.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proc. of NAACL*.

Qi, Y., Sachan, D., Felix, M., Padmanabhan, S., and Neubig, G. (2018). When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana, June. Association for Computational Linguistics.

Radford, A. (2018). Improving language understanding by generative pre-training.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2019). Exploring the limits of transfer learning with a unified text-to-text transformer.

Ren, S., Wu, Y., Liu, S., Zhou, M., and Ma, S. (2019). Explicit cross-lingual pre-training for unsupervised machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 770–779, Hong Kong, China, November. Association for Computational Linguistics.

Sennrich, R. and Haddow, B. (2016). Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany, August. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016a). Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016b). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August. Association for Computational Linguistics.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. (2019). Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th Neural Information Processing Systems Conference (NIPS)*, pages 3104–3112, Montréal, Canada.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 30th Neural Information Processing Systems Conference (NIPS)*, pages 5998–6008, Long Beach, USA.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Zhang, J. and Zong, C. (2016). Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas, November. Association for Computational Linguistics.

Zhou, C., Ma, X., Hu, J., and Neubig, G. (2019). Handling syntactic divergence in low-resource machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1388–1394, Hong Kong, China, November. Association for Computational Linguistics.

Zoph, B., Yuret, D., May, J., and Knight, K. (2016). Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1568–1575, Austin, USA.