UnihanLM: Coarse-to-Fine Chinese-Japanese Language Model Pretraining with the Unihan Database

Canwen Xu[1]*, Tao Ge[2], Chenliang Li[3], Furu Wei[2]
[1] University of California, San Diego [2] Microsoft Research Asia [3] Wuhan University
[1] cxu@ucsd.edu [2] {tage,fuwei}@microsoft.com [3] cllee@whu.edu.cn

Abstract

Chinese and Japanese share many characters with similar surface morphology. To better utilize the shared knowledge across the languages, we propose UnihanLM, a self-supervised Chinese-Japanese pretrained masked language model (MLM) with a novel two-stage coarse-to-fine training approach. We exploit Unihan, a ready-made database constructed by linguistic experts to first merge morphologically similar characters into clusters. The resulting clusters are used to replace the original characters in sentences for the coarse-grained pretraining of the MLM. Then, we restore the clusters back to the original characters in sentences for the fine-grained pretraining to learn the representation of the specific characters. We conduct extensive experiments on a variety of Chinese and Japanese NLP benchmarks, showing that our proposed UnihanLM is effective on both mono- and cross-lingual Chinese and Japanese tasks, shedding light on a new path to exploit the homology of languages.[1]

1 Introduction

Recently, Pretrained Language Models have shown promising performance on many NLP tasks (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019c; Lan et al., 2020). Many attempts have been made to train a model that supports multiple languages. Among them, Multilingual BERT (mBERT) (Devlin et al., 2019) is released as a part of BERT. It directly adopts the same model architecture and training objective, and is trained on Wikipedia in different languages. XLM (Lample and Conneau, 2019) is proposed with an additional language embedding and a new training objective (translation language modeling, TLM). XLM-R (Conneau et al., 2019) has a larger size and is trained with more data. Based on XLM, Unicoder (Huang et al., 2019) collects more data and uses multi-task learning to train on three supervised tasks.

JA	台^{1}風^{2}は熱^{3}帯^{4}低気^{5}圧^{6}の一種^{7}です。
T-ZH	颱^{1}風^{2}是熱^{3}帶^{4}氣^{5}旋的一種^{7}。
S-ZH	台^{1}风^{2}是热^{3}带^{4}低气^{5}压^{6}的一种^{7}。
EN	Typhoon is a type of tropical depression.

Table 1: A sentence example in Japanese (JA), Traditional Chinese (T-ZH) and Simplified Chinese (S-ZH) with its English translation (EN). The characters that already share the same Unicode are marked with an underline. In this work, we further match characters with identical meanings but different Unicode, then merge them. Characters eligible to be merged together are marked with the same superscript.

The census of cross-lingual approaches is to allow lexical information to be shared between languages. XLM and mBERT exploit shared lexical information by Byte Pair Encoding (BPE) (Sennrich et al., 2016) and WordPiece (Wu et al., 2016), respectively. However, these automatically learned shared representations have been criticized by recent work (K et al., 2020), which reveals their limitations in sharing meaningful semantics across languages. Also, words in both Chinese and Japanese are short, which prohibits an effective learning of sub-word representations. Different from European languages, Chinese and Japanese naturally share Chinese characters as a subword component. Early work (Chu et al., 2013) shows that shared characters in these two languages can benefit Example-based Machine Translation (EBMT) with a statistical based phrase extraction and alignment. For Neural Machine Translation (NMT), (Zhang and Ko-

[*] This work was done during Canwen's internship at Microsoft Research Asia.

[1] The code and pretrained weights are available at https://github.com/JetRunner/unihan-lm.

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 201–211
December 4 - 7, 2020. ©2020 Association for Computational Linguistics

machi, 2019) exploited such information by learning a BPE representation over sub-character (i.e., ideograph and stroke) sequence. They applied this technique to unsupervised Chinese-Japanese machine translation and achieved state-of-the-art performance. However, this approach greatly relies on unreliable automatic BPE learning and may suffer from the noise brought by various variants.

To facilitate lexical sharing, we propose **Unihan Language Model** (UnihanLM), a cross-lingual pretrained masked language model for Chinese and Japanese. We propose a two-stage coarse-to-fine pretraining procedure to empower better generalization and take advantages of shared characters in Japanese, Traditional and Simplified Chinese. First, we let the model exploit maximum possible shared lexical information. Instead of learning a shared sub-word vocabulary like the prior work, we leverage Unihan database (Jenkins et al., 2019), a ready-made constituent of the Unicode standard, to extract the shared lexical information across the languages. By exploiting this database, we can effectively merge characters with the similar surface morphology but independent Unicodes, as shown in Table 1 into thousands of clusters. The clusters will be used to replace the characters in sentences during the first-stage coarse-grained pretraining. After the coarse-grained pretraining finishes, we restore the clusters back to the original characters and initialize their representation with their corresponding cluster's representation and then learn their specific representation during the second-stage fine-grained pretraining. In this way, our model can make full use of shared characters while maintaining a good sense for nuances of similar characters.

To verify the effectiveness of our approach, we evaluate on both lexical and semantic tasks in Chinese and Japanese. On word segmentation, our model outperforms monolingual and multilingual BERT (Devlin et al., 2019) and shows a much higher performance on cross-lingual zero-shot transfer. Also, our model achieves state-of-the-art performance on unsupervised Chinese-Japanese machine translation, and is even comparable to the supervised baseline on Chinese-to-Japanese translation. On classification tasks, our model achieves a comparable performance with monolingual BERT and other cross-lingual models trained with the same scale of data.

To summarize, our contributions are three-fold: (1) We propose UnihanLM, a cross-lingual pretrained language model for Chinese and Japanese NLP tasks. (2) We pioneer to apply the language resource – the Unihan Database to help model pretraining, allowing more lexical information to be shared between the two languages. (3) We devise a novel coarse-to-fine two-stage pretraining strategy with different granularity for Chinese-Japanese language modeling.

2 Preliminaries

2.1 Chinese Character

Chinese character is a pictograph used in Chinese and Japanese. These characters often share the same background and origin. However, due to historic reasons, Chinese characters have developed into different writing systems, including Japanese Kanji, Traditional Chinese and Simplified Chinese. Also, even in a single text, multiple variants of the same characters can be used interchangeably (e.g., "台灣" and "臺灣" for "Taiwan", in Traditional Chinese). These characters have identical or overlapping meanings. Thus, it is critical to better exploit such information for modeling both cross-lingual (i.e., between Chinese and Japanese), cross-system (i.e., between Traditional and Simplified Chinese) and cross-variant semantics.

Both Chinese and Japanese have no delimiter (e.g., white space) to mark the boundaries of words. There have always been debates over whether word segmentation is necessary for Chinese NLP. Recent work (Li et al., 2019) concludes that it is not necessary for various NLP tasks in Chinese. Previous cross-lingual language models use different methods for tokenization. mBERT adds white spaces around Chinese characters and lefts Katakana/Hiragana Japanese (also known as kanas) unprocessed. Different from mBERT, XLM uses Stanford Tokenizer[2] and KyTea[3] to segment Chinese and Japanese sentences, respectively. After tokenization, mBERT and XLM use WordPiece (Wu et al., 2016) and Byte Pair Encoding (Sennrich et al., 2016) for sub-word encoding, respectively.

Nevertheless, both approaches suffer from obvious drawbacks. For mBERT, the kanas and Chinese characters are treated differently, which causes a mismatch for labeling tasks. Also, leaving kanas untokenized may cause the data sparsity problem. For XLM, as pointed out in (Li et al., 2019), an

[2] https://nlp.stanford.edu/software/tokenizer.html

[3] http://www.phontron.com/kytea/

Variant	Description	Example
Traditional Variant	The traditional versions of a simplified Chinese character.	发 → 髮 (hair), 發 (to burgeon)
Simplified Variant	The simplified version of a traditional Chinese character.	團 → 团 (group)
Z-Variant	Same character with different unicodes only for compatibility.	説 ↔ 說 (say)
Semantic Variant	Characters with identical meaning.	兎 ↔ 兔 (rabbit)
Specialized Semantic Variant	Characters with overlapping meaning.	丼 (rice bowl, well) ↔ 井 (well)

Table 2: The five types of variants in the Unihan database.

external word segmenter would introduce extra segmentation errors and compromise the performance of the model. Also, as a word-based model, it is difficult to share cross-lingual characters unless the segmented words in both Chinese and Japanese are exactly matched. Furthermore, both approaches would enlarge the vocabulary size and thus introduce more parameters.

2.2 Unihan Database

Chinese, Japanese and Korean (CJK) characters share a common origin from the ancient Chinese characters. However, with the development of each language, both the shape and semantics of characters drastically change. When exchanging information, different codings of the same character hinders the text processing. Thus, as the result of Han unification[4], the database of CJK Unified Ideographs, Unihan (Jenkins et al., 2019), is constructed by human experts tracing the sources of each character.

As part of the Unicode Standard, Unihan merges the Unicode for some characters from different languages and provides extra variant information between different characters. In previous studies (Zhang and Komachi, 2019; Lample and Conneau, 2019; Devlin et al., 2019), Unicode is used by default. However, due to the "Source Separation Rule" of Unicode, to remain the compatibility with prior encoding systems, a single character can have multiple Unicodes with different glyphs. For example, for the character "戸", there are three unicodes: U+6236, U+6237 and U+6238. This feature could be useful for message exchange but is undoubtedly undesirable for NLP and may bring the problems of data sparsity and prevent the alignment of a cross-lingual language model.

Fortunately, Unihan database also provides 12,373 entries of variant information in five types, as listed in Table 2. Note that one character may have multiple types of variants and each type may have multiple variant characters (e.g., the traditional variants of "发" in Table 2). Such information forms a complex graph structure.

Tokenization Scheme	Result
BERT (2019)	台風 / はひどい
XLM (2019)	台風 / は / ひどい
UnihanLM	台 / 風 / は / ひ / ど / い

Table 3: Different tokenization schemes used in recent work and ours. Note that the tokenized results of both BERT and XLM in this table are before WordPiece/BPE applied. WordPiece/BPE may further split a token.

3 UnihanLM

In this section, we introduce the tokenization, character merging and training procedure for our proposed UnihanLM.

3.1 Tokenization

As analyzed in Section 2.1, the tokenization scheme is tricky and critical for East Asian languages. Although recent work (Li et al., 2019) reveals that tokenization is unnecessary for most high-level NLU and NLG tasks, many downstream labeling tasks (e.g., Part-of-speech Tagging, Named Entity Recognition) still require an implicit or explicit segmentation. To enable all NLP tasks, we tokenize the sentences by treating every character (including Japanese Kana) as a token. Thus, our model is capable of processing all tasks, from the lowest-level Chinese and Japanese word segmentation to high-level NLU tasks. We summarize the different tokenization schemes used in recent work and ours in Table 3.

We do not further apply BPE to our tokenized sentences for two reasons. First, a character is the atomic element in both Chinese and Japanese grammars which should not be further split. Second, character itself is naturally a sub-word semantic element, e.g., "自" (self) + "信"(belief) = "自信"

[4] https://en.wikipedia.org/wiki/Han_unification

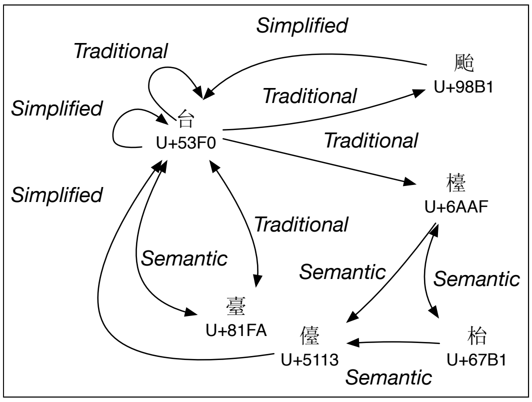

Figure 1: A connected subgraph of Unihan database. For example, for the word “typhoon”, “台” is used in Japanese and Simplified Chinese while “颱” is used in Traditional Chinese.

(confidence); “自” (self) + “尊”(respect) = “自尊” (self-esteem).

3.2 Character Merging

To reduce the vocabulary size and align the Chinese characters in Traditional Chinese, Simplified Chinese and Japanese to the greatest extent, it is important to merge as many characters as possible while ensuring only merging characters with the identical or overlapping meanings. Thus, we use Unihan database, which includes character variant information collected and approved by human experts. We use four types of variants including Traditional Variant, Simplified Variant, Z-Variant and Semantic Variant. Note that we exclude Specialized Semantic Variant which may raise ambiguity problem since it is not very common and the semantics of the two characters are merely overlapping, not identical.

However, merging characters is still challenging since the variant information in Unihan database is a complex graph, as illustrated in Figure 1. To merge the characters as much as possible, we convert Unihan database to a large undirected graph and use Union Find Algorithm (Galler and Fischer, 1964) to find all maximal connected subgraph. For example, the whole Figure 1 is a subgraph in the Unihan graph found by the algorithm. We call all characters in a maximal connected subgraph belong to a “cluster”. After this merging procedure, the 12,373 variant entries yield a total of 4,001 clusters.

3.3 Training Procedure

As illustrated in Figure 2, the model is a Transformer based model with three embeddings as input and the training procedure is composed of two phases.

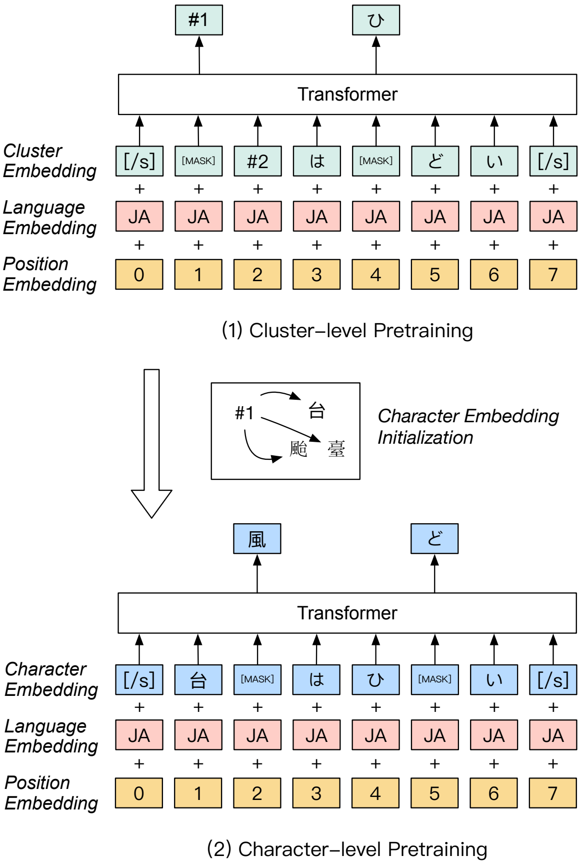

Figure 2: The model architecture of UnihanLM. (1) We merge characters to clusters and use cluster indices when doing cluster-level pretraining. In the figure, “#1” and “#2” indicate indices of the clusters which “台” and “風” belong to, respectively. (2) We initialize the embedding of each character in a cluster with the cluster embedding and do character-level pretraining to predict each character.

3.3.1 Model

Our model is a Transformer-based Masked Language Model (Devlin et al., 2019) which learns to predict the randomly masked words with the context. Also, following (Lample and Conneau, 2019), we add language embedding to help the model distinguish between Chinese and Japanese, especially when we share the characters between these two languages. The detailed hyperparameter settings are described in Section 4.1.

3.3.2 Coarse-grained Cluster-level Pretraining

To maximize the shared lexicon and force them to share a representation, we leverage clusters to pretrain our models on a coarse-grained cluster level.

We first append the cluster indices to the vocabulary. During cluster-level pretraining, we substitute the character index with its corresponding cluster index if the character is in the Unihan database. For Japanese kanas, punctuation, number and other characters not in Unihan database, we keep its original token index. In this way, we employ human prior knowledge to the pretraining procedure and allow the model to roughly model the semantic knowledge.

3.3.3 Fine-grained Character-level Pretraining

Although the clusters training is effective, there are two problems remaining unsolved. First, Traditional Variant could be ambiguous. As shown in Table 2, a character (most likely one used in Simplified Chinese) may have multiple Traditional Variants. Although it should not have a significant negative effect for understanding the language (since a Simplified Chinese user can disambiguate between different meanings of a character based on its context), it still makes sense to improve the overall performance by distinguish the characters explicitly (Navigli et al., 2017). Also, in tasks involving decoding (e.g., machine translation), they must be processed independently. Thus, character disambiguation can be naturally used as a self-supervised task. Second, when using the trained model for translation, it would be important for the model to decode the right character for different languages and writing systems. For example, for the word meaning "typhoon", "台風", "颱風", "台风" should be used in Japanese, Traditional Chinese and Simplified Chinese, respectively.

Consequently, we leave these nuances of characters to a fine-grained character-level pretraining. Since during the cluster-level pretraining, all characters in Unihan database are preserved in the vocabulary but their embedding is untrained, we initialize their embedding with their corresponding cluster embedding trained in cluster-level pretraining stage. In the character-level pretraining stage, we discard the clusters in the vocabulary and do not substitute any character since then. In this way, the model can handle each character case by case, with a fine granularity. We restart the training with a smaller learning rate to allow the model to learn to disambiguate.

Model	#Layer	#Param.
BERT-Mono-ZH (2019)	12	110M
mBERT (2019)	12	179M
XLM (2019)	16	571M
UnihanLM	12	176M

Table 4: The numbers of layers and parameters for different models.

4 Experiments

In this section, we compare UnihanLM with other self-supervised pretrained language models. All of our baselines (monolingual BERT, mBERT and XLM) use Wikipedia for self-supervised pretraining. Note that we do not compare our model to XLM-R (Conneau et al., 2019) and Unicoder (Huang et al., 2019) since they are trained with much more data and even on supervised tasks.

4.1 Training Details

We use the mixture of Chinese and Japanese Wikipedia[5] as the unparalleled pretraining corpus. We sample $5,000$ sentences as validation set for model selection and use the rest for training. Our model uses 12 layers of Transformer blocks with 16 attention heads. The hidden size is set to 1,024. The vocabulary size is 24,044. Shown in Table 4, our model has a similar size to mBERT. We train our model on 8 Nvidia V100 32GB GPUs to optimize Masked Language Model (MLM) objective (Devlin et al., 2019) with an Adam (Kingma and Ba, 2015) optimizer. The masking probability is set to 15%. We add a L2 regularization of 0.01. We warm up the first 30,000 steps for each stage of pretraining by an inverse square root function. The batch size is set to 64 per GPU. The maximum sequence length is limited to 256 tokens. We add dropout (Srivastava et al., 2014) for both feed-forward network and attention with a drop rate of 0.1. The learning rate for cluster-level pretraining is set to 1×10^{-4}. After 264 hours of cluster-level pretraining until convergence, we perform character-level pretraining with a smaller learning rate of 5×10^{-5} for another 43 hours. We choose the best model according to its perplexity on validation set. For downstream tasks (to be detailed shortly), we fine-tune UnihanLM with a learning rate of 5×10^{-7}, 1×10^{-4}, 2.5×10^{-5} and a batch

[5] https://dumps.wikimedia.org/

Method	PKU (ZH)	KWDLC (JA)
Standard training		
mBERT (2019)	95.0	96.3
BERT-Mono-ZH (2019)	96.5	-
UnihanLM	**96.6**	**98.2**
Cross-lingual zero-shot transfer		
mBERT (2019)	82.0	63.1
UnihanLM	**85.7**	**74.1**

Table 5: F1 scores on Chinese Word Segmentation (CWS) and Japanese Word Segmentation (JWS) tasks. "Cross-lingual zero-shot transfer" indicates that the model is trained on CWS and zero-shot tested on JWS, vice versa.

size of 20, 24, 16 for word segmentation, unsupervised machine translation and classification tasks, respectively.

4.2 Word Segmentation

Word segmentation is a fundamental task in both Chinese and Japanese NLP. It is often recognized as the first step for further processing in many systems. Thus, we evaluate Chinese Word Segmentation (CWS) and Japanese Word Segmentation (JWS) on PKU dataset (Emerson, 2005) and KWDLC (Kawahara et al., 2014). We use Multilingual BERT and monolingual Chinese BERT (Devlin et al., 2019) as baselines. We use pretrained checkpoints provided by Google[6]. Following previous work, we treat the word segmentation task as a sequence labeling task. Note that XLM (Lample and Conneau, 2019) uses pre-segmented sentences as input, making it inapplicable for this task. As shown in Table 5, our proposed UnihanLM outperforms mBERT and monolingual BERT by 1.6 and 0.1 in terms of F1 score on CWS, respectively. On JWS, our model outperforms mBERT by 1.9 on F1. Additionally, we conduct zero-shot transfer experiments to determine how much lexical knowledge is shared within Chinese and Japanese for each model. We use the weights trained on CWS and JWS for zero-shot transferring on the other language. Our model drastically outperforms mBERT on this task by 3.7 and 11.0 on CWS and JWS, respectively. This proves that our model can better capture the lexical knowledge shared between Chinese and Japanese. Also, it is notable that zero-shot JWS has a prominently poorer performance than zero-shot CWS. As we analyze, the criterion for segmenting Chinese characters can be learned with a Japanese corpus and then transferred to CWS. However, since no kana is present in CWS, the model cannot successfully segment kanas, when performing zero-shot inference on JWS.

Method	ZH→JA	JA→ZH
Supervised baseline		
OpenNMT (Klein et al., 2017)	42.12	40.63
Fine-tuned on Wikipedia		
XLM (Lample and Conneau, 2019)	14.58	15.06
UnihanLM	**33.53**	**28.70**
Fine-tuned on shuffled ASPEC-JC training set		
Stroke (Zhang and Komachi, 2019)	33.81	31.66
UnihanLM	**44.59**	**40.58**

Table 6: BLEU scores of Chinese-Japanese unsupervised translation on ASPEC-JC dataset.

4.3 Unsupervised Machine Translation

A Chinese speaker who never learned Japanese can roughly understand a Japanese text (and vice versa), due to the similarity between the writing systems of these two languages. On the other hand, only a few parallel corpora between Chinese and Japanese are publicly available, and they are usually small in size. Thus, Unsupervised Machine Translation (UMT) is very promising and meaningful on the Chinese-Japanese translation task. We evaluate on Asian Scientific Paper Excerpt Corpus Japanese-Chinese (ASPEC-JC)[7], the most widely-used Chinese-Japanese Machine Translation dataset. We perform our experiments under two settings: (1) Chinese and Japanese Wikipedia is used as the monolingual corpora, following the setting of (Lample and Conneau, 2019). (2) Shuffled unparalleled ASPEC-JC training set is used as the monolingual corpora, following the settings in (Zhang and Komachi, 2019).

Except for XLM, we choose (Zhang and Komachi, 2019), the current state-of-the-art Chinese-Japanese UMT model as a strong baseline. They decomposed a Chinese character in both Chinese and Japanese into strokes and then learn a shared token in the stroke sequence to increase the shared tokens in the vocabulary. However, this method relies on an unsupervised BPE (Sennrich et al., 2016) to learn shared stroke tokens from a long noisy

[6] https://github.com/google-research/bert

[7] http://orchid.kuee.kyoto-u.ac.jp/ASPEC/

Method	PAWS-X ZH	PAWS-X JA
BOW	54.5	55.1
ESIM (Chen et al., 2017a)	60.3	59.6
mBERT (Devlin et al., 2019)	82.3	79.2
XLM (Lample and Conneau, 2019)	82.5	79.5
UnihanLM	**82.7**	**80.5**

Table 7: Accuracy scores on PAWS-X dataset.

stroke sequence, which is rather unreliable compared to our solution. For example, “丑” (ugly) and “五” (five) have a very similar stroke sequence but completely different meanings. Following (Lample and Conneau, 2019), we use our pretrained weights to initialize the translation model and train the model with denoising auto-encoding loss and online back-translation loss. Note that both baselines use Wikipedia as the unsupervised data and are based on the same UMT method (Lample et al., 2018c). We use character-level BLEU (Papineni et al., 2002) as the evaluation metric.

We demonstrate the results in Table 6. As we analyzed, XLM suffers from a severe out-of-vocabulary (OOV) problem on AESPEC-JC, a dataset composed of scientific papers, containing many new terminologies which do not show up in the pretraining corpus of XLM. As a word-based model, XLM is not able to handle these new words and thus yields a rather poor result. When fine-tuned on unparalleled training set of ASPEC-JC, our model outperforms the previous state-of-the-art model (Zhang and Komachi, 2019) by a large margin of 10.78 and 8.92 in terms of BLEU. Also notably, UnihanLM even outperforms the supervised baseline on Chinese-to-Japanese translation and has a performance in close proximity on Japanese-to-Chinese task, compared to an early supervised machine translation model, OpenNMT (Klein et al., 2017), trained on the paired training set of ASPEC-JC.

4.4 Text Classification

To further evaluate our model, we perform our experiments on Cross-lingual Paraphrase Aversaries from Word Scrambling (PAWS-X) (Yang et al., 2019b), a newly proposed cross-lingual text classification dataset supporting seven languages including Chinese and Japanese. This dataset consists of challenging English paraphrase identification pairs from Wikipedia and Quora. Then the human translators translate the text into the other six languages. We test under the setting of *TRANSLATE-TRAIN* (i.e., we use the provided translation of the training set for both Chinese and Japanese and test in the same language). Shown in Table 7, UnihanLM outperforms all baselines in (Yang et al., 2019b), including mBERT.

4.5 Ablation Study

To verify the effectiveness of our two-stage pretraining procedure, we conduct an ablation study. A character-level model is trained from scratch without the cluster-level pretraining and marked as “$-cluster$”. On the other hand, we use the model trained in cluster-level stage for downstream tasks and mark it as “$-character$”. Note that since the objective for cluster-level stage is to predict the masked cluster, it cannot be used for unsupervised translation. Shown in Figure 8, both cluster-level and character-level pretraining play an essential role on classification tasks. On translation task, cluster-level pretraining is more important when fine-tuned on Wikipedia but has a relatively smaller impact when using shuffled ASPEC-JC training set.

To analyze the success of our two-stage training strategy, we would like to emphasize two strengths. First, as mentioned before, our easy-to-hard training procedure matches the core idea of Curriculum Learning (Bengio et al., 2009), which smooths the training and help the model generalize better. Second, the two-stage procedure inherently introduces a new self-supervised task, which could take the advantage of Multitask Learning (Caruana, 1993).

5 Related Work

Multilingual Representation Learning Learning cross-lingual representations are useful for downstream tasks such as cross-lingual classification (Conneau et al., 2018; Yang et al., 2019b), cross-lingual retrieval (Zweigenbaum et al., 2017; Artetxe and Schwenk, 2019) and cross-lingual QA (Artetxe et al., 2019; Lewis et al., 2019; Clark et al., 2020). Earlier work on multilingual representations exploiting parallel corpora (Luong et al., 2015; Gouws et al., 2015) or a bilingual dictionary to learn a linear mapping (Mikolov et al., 2013; Faruqui and Dyer, 2014). Subsequent methods explored self-training (Artetxe et al., 2017) and unsupervised learning (Zhang et al., 2017; Artetxe et al., 2018; Lample et al., 2018b). Recently, multilingual pretrained encoders have shown its effec-

Method	PAWS-X		ASPEC-JC			
			Wiki		Shuffled-train	
	ZH	JA	ZH→JA	JA→ZH	ZH→JA	JA→ZH
UnihanLM	**82.7**	**80.5**	**33.53**	**28.70**	**44.59**	**40.58**
−cluster	81.5	79.2	29.33	20.93	42.34	39.24
−character	82.0	80.1	-	-	-	-

Table 8: The results of ablation study on text classification and UMT. "-cluster" and "-character" indicate the model trained without the cluster-level pretraining and character-level pretraining, respectively. The metrics for PAWS-X and ASPEC-JC are accuracy and BLEU, respectively.

tiveness for learning deep cross-lingual representations (Eriguchi et al., 2018; Pires et al., 2019; Wu and Dredze, 2019; Lample and Conneau, 2019; Conneau et al., 2019; Huang et al., 2019).

Word Segmentation Word segmentation is often formalized as a sequence tagging task. It requires lexical knowledge to split a character sequence into a word list that can be used for downstream tasks. This step is necessary for many earlier NLP systems for Chinese and Japanese. Recent work on Chinese Word Segmentation (Wang and Xu, 2017; Zhou et al., 2017; Yang et al., 2017; Cai et al., 2017; Chen et al., 2017b; Yang et al., 2019a) and Japanese Word Segmentation (Kaji and Kitsuregawa, 2014; Fujinuma and II, 2017; Kitagawa and Komachi, 2018) exploit deep neural networks and focus on building end-to-end sequence tagging models.

Unsupervised Machine Translation Recently, machine translation systems have demonstrated near human-level performance on some languages. However, it depends on the availability of large amounts of parallel sentences. Unsupervised Machine Translation addresses this problem by exploiting monolingual corpora which can be easily constructed. Lample et al. (2018a) proposed a UMT model by learning to reconstruct in both languages from a shared feature space. Lample et al. (2018c) exploited language modeling and back-translation and thus proposed a neural unsupervised translation model and a phase-based translation model. Different from European languages (e.g., English), Chinese and Japanese naturally share Chinese characters. Zhang and Komachi (2019) exploited such information by learning a BPE representation over sub-character (i.e., ideograph and stroke) sequence. They applied this technique to unsupervised Chinese-Japanese machine translation and achieved state-of-the-art performance. This information is also shown to be effective by (Xu et al., 2019).

6 Discussion and Future Work

There is still space to improve for our method. First, as we analyze, except for Chinese characters, English words often appear in both Chinese and Japanese texts. In our current model, they are treated as normal characters without any special processing. However, such a rough processing may harm the performance of the model on some tasks. For example, in PAWS-X, many entities remain untranslated and this may have a negative effect on the performance of our model. Also, loan words (i.e., Gairaigo), especially from English, constitute a large part of nouns in modern Japanese (Miller, 1998). These words are written with kanas, instead of Chinese characters which makes it inapplicable to be shared with our approach. Thus, it may be reasonable to involve English in cross-lingual modeling of Asian languages, as well. Similarly, Chinese characters exist in Korean and Vietnamese but are now written in Hangul (Korean alphabet) and Vietnamese alphabet, respectively. Our future work will explore the possibility to generalize the idea to more Asian languages including Korean and Vietnamese.

7 Conclusion

In this paper, we exploit the ready-made Unihan database constructed by linguistic experts and propose a novel Chinese-Japanese cross-lingual language model trained by a two-stage coarse-to-fine procedure. Our extensive experiments on word segmentation, unsupervised machine translation and text classification verify the effectiveness of our model. Our approach sheds some light on the linguistic features that receive insufficient attention recently and showcases a novel way to fuse human linguistic knowledge and exploit the similarity between two languages.

Acknowledgments

We are grateful for the insightful comments from the anonymous reviewers. We would like to thank Longtu Zhang and Mamoru Komachi from Tokyo Metropolitan University for their help with the MT baseline.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *ACL*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *ACL*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2019. On the cross-lingual transferability of monolingual representations. *CoRR*, abs/1910.11856.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Trans. Assoc. Comput. Linguistics*, 7:597–610.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*.

Deng Cai, Hai Zhao, Zhisong Zhang, Yuan Xin, Yongjian Wu, and Feiyue Huang. 2017. Fast and accurate neural word segmentation for chinese. In *ACL*.

Rich Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias. In *ICML*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017a. Enhanced LSTM for natural language inference. In *ACL*.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017b. Adversarial multi-criteria learning for chinese word segmentation. In *ACL*.

Chenhui Chu, Toshiaki Nakazawa, Daisuke Kawahara, and Sadao Kurohashi. 2013. Chinese-japanese machine translation exploiting chinese characters. *ACM Trans. Asian Lang. Inf. Process.*, 12(4):16:1–16:25.

Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi QA: A benchmark for information-seeking question answering in typologically diverse languages. *CoRR*, abs/2003.05002.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: evaluating cross-lingual sentence representations. In *EMNLP*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Thomas Emerson. 2005. The second international chinese word segmentation bakeoff. In *SIGHAN@IJCNLP*.

Akiko Eriguchi, Melvin Johnson, Orhan Firat, Hideto Kazawa, and Wolfgang Macherey. 2018. Zero-shot cross-lingual classification using multilingual neural machine translation. *CoRR*, abs/1809.04686.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *EACL*.

Yoshinari Fujinuma and Alvin Grissom II. 2017. Substring frequency features for segmentation of japanese katakana words with unlabeled corpora. In *IJCNLP*.

Bernard A. Galler and Michael J. Fischer. 1964. An improved equivalence algorithm. *Commun. ACM*, 7(5):301–303.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *ICML*.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. In *EMNLP-IJCNLP*.

John H. Jenkins, Richard Cook, and Ken Lunde. 2019. Uax #38: Unicode han database (unihan). `http://www.unicode.org/reports/tr38/tr38-27.html`.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: An empirical study. In *ICLR*.

Nobuhiro Kaji and Masaru Kitsuregawa. 2014. Accurate word segmentation and POS tagging for japanese microblogs: Corpus annotation and joint modeling with lexical normalization. In *EMNLP*. ACL.

Daisuke Kawahara, Yuichiro Machida, Tomohide Shibata, Sadao Kurohashi, Hayato Kobayashi, and Manabu Sassano. 2014. Rapid development of a corpus with discourse annotations using two-stage crowdsourcing. In *COLING*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Yoshiaki Kitagawa and Mamoru Komachi. 2018. Long short-term memory for japanese word segmentation. In *PACLIC*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *ACL*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *NeurIPS*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *ICLR*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018b. Word translation without parallel data. In *ICLR*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018c. Phrase-based & neural unsupervised machine translation. In *EMNLP*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *ICLR*. OpenReview.net.

Patrick S. H. Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2019. MLQA: evaluating cross-lingual extractive question answering. *CoRR*, abs/1910.07475.

Xiaoya Li, Yuxian Meng, Xiaofei Sun, Qinghong Han, Arianna Yuan, and Jiwei Li. 2019. Is word segmentation necessary for deep learning of chinese representations? In *ACL*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *VS@HLT-NAACL*.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Laura Miller. 1998. Wasei eigo: English "loanwords" coined in japan. *The life of language: Papers in linguistics in honor of William Bright*, pages 123–139.

Roberto Navigli, José Camacho-Collados, and Alessandro Raganato. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *EACL*.

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

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? In *ACL*. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Chunqi Wang and Bo Xu. 2017. Convolutional neural network with word embeddings for chinese word segmentation. In *IJCNLP*.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *EMNLP-IJCNLP*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Canwen Xu, Feiyang Wang, Jialong Han, and Chenliang Li. 2019. Exploiting multiple embeddings for chinese named entity recognition. In *CIKM*.

Jie Yang, Yue Zhang, and Fei Dong. 2017. Neural word segmentation with rich pretraining. In *ACL*.

Jie Yang, Yue Zhang, and Shuailong Liang. 2019a. Subword encoding in lattice LSTM for chinese word segmentation. In *NAACL-HLT*.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019b. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *EMNLP-IJCNLP*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019c. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Longtu Zhang and Mamoru Komachi. 2019. Chinese-japanese unsupervised neural machine translation using sub-character level information. In *The 33rd Pacific Asia Conference on Language, Information and Computation*.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *EMNLP*.

Hao Zhou, Zhenting Yu, Yue Zhang, Shujian Huang, Xin-Yu Dai, and Jiajun Chen. 2017. Word-context character embeddings for chinese word segmentation. In *EMNLP*.

Pierre Zweigenbaum, Serge Sharoff, and Reinhard Rapp. 2017. Overview of the second BUCC shared task: Spotting parallel sentences in comparable corpora. In *BUCC@ACL*.