An Unsupervised Framework for Adaptive Context-aware Simplified-Traditional Chinese Character Conversion

Wei Li†, Shutan Huang†, Yanqiu Shao
School of Information Science, Beijing Language and Culture University
Xueyuan Road 15th, Haidian District, Beijing, China
liweitj47@blcu.edu.cn, shutan2022@163.com, shaoyanqiu@blcu.edu.cn

Abstract

Traditional Chinese character is an important carrier of Chinese culture, and is still actively used in many areas. Automatic conversion between traditional and simplified Chinese characters can help modern people understand traditional culture and facilitate communication among different regions. Previous conversion methods rely on rule-based mapping or shallow feature-based machine learning models, which struggle to convert simplified characters with different origins and constructing training data is costly. In this study, we propose an unsupervised adaptive context-aware conversion model that learns to convert between simplified and traditional Chinese characters under a denoising auto-encoder framework **requiring no labeled data**. Our model includes a **Latent Generative Adversarial Encoder** that transforms vectors to a latent space with generative adversarial network, which adds noise as an inevitable side effect, Based on which a **Context-aware Semantic Reconstruction Decoder** restores the original input while considering a broader range of context with a pretrained language model. Additionally, we propose to apply **early exit mechanism** during inference to reduce the computation complexity and improve the generalization ability. To test the effectiveness of our model, we construct a high quality test dataset with simplified-traditional Chinese character text pairs. Experiment results and extensive analysis demonstrate that our model outperforms strong unsupervised baselines and yields better conversion result for one-to-many cases.

Keywords: Simplified-traditional Chinese Character Conversion, Unsupervised Learning Framework, Multi-to-one Mapping Problem

1. Introduction

Traditional Chinese characters have long been used in the history of China, and are still being actively applied in regions like Hong Kong and Taiwan. As a result of their widespread use, traditional Chinese characters play a crucial role in preserving historical knowledge and culture of China. Despite its importance, due to the adoption of simplified Chinese characters, many individuals in the mainland China struggle with reading and writing traditional Chinese characters. This difference in the writing system not only hinders the communication among different regions, but also brings difficulty for the modern Chinese in understanding their ancient culture. Therefore, it is of great significance to automatically convert between traditional Chinese characters and simplified Chinese characters.

Traditional converting methods rely on mapping tables (Li et al., 2010). Such methods work for many of the characters. However, the simplification process of Chinese characters not only simplified the writing pattern, but also reduced the vocabulary size by merging different characters to a single one, creating ambiguity in the character system, which makes mapping-based methods ineffective when dealing with one-to-many conversion problems. To eliminate the ambiguity, Chen et al. (2011) propose to apply log-linear model with human engineered features. Although considers the context around the target characters, it is heavily reliant on the quality of human-crafted features and can only model shallow semantics within a limited context. Furthermore, compared with the traditional mapping based methods, such supervised methods require large amount of labeled training data to cover as many as Chinese characters, which is costly to build and may render the model vulnerable to out-of-vocabulary problems.

To make the task more difficult, the documents cover a long period of time, ranging from ancient to modern times, resulting in a diverse range of expression patterns. Therefore, it is essential for the model to be adaptive to the targeted documents from different times. Furthermore, due to the variance of expression forms, it is expensive and impractical to manually align text pairs of traditional and simplified Chinese characters across different ages, making self-supervised methods an attractive alternative. However, we observe that luckily many characters remain the same after simplification and the character number (sentence length) within the source text is also the same. These features make this task more plausible to apply unsupervised methods.

Inspired by the success of pretrained language models (PLM), we propose to introduce PLM to

[1] †Equal Contribution

LREC-COLING 2024, pages 1318–1326
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

model the semantic context of targeted characters for character disambiguation, which is powerful for long range dependency. To liberate the model from the need for parallel labeled data and enhance its adaptability to diverse target texts, we propose an unsupervised learning model under a denoising auto-encoder framework. Our model takes **monolingual** text in characters (either simplified or traditional) as input, which are encoded and mapped into the representation space of the other character set (e.g., simplified as input, mapped into traditional space) with Latent Generative Adversarial Encoder. The model is then asked with restoring the original text from the mapped hidden states, which finishes the loop of auto-encoding. Due to the unsupervised nature of the encoding process, noise is inevitably introduced, which makes the entire training loop equivalent to denoising from polluted representations. Based on the observation that different characters require varying levels of contextual information (e.g., many characters remain the same after simplification), we propose to incorporate early exit mechanism to optimize the computation complexity considering the needs of each character, which also enhances the generalization of the model.

To test the effectiveness of our proposed method, we collect high-quality text pairs of traditional and simplified Chinese characters from Chinese Text Project[1] as the test set. The experimental results show that our model outperforms all existing publicly available software for simplified-traditional Chinese character conversion. Extensive analysis demonstrates that our proposed early exit mechanism can not only reduce the theoretic computation complexity, but also improve the prediction accuracy because of its better generalization ability. [2]

We conclude our contributions as follows,

- We propose an **unsupervised adaptive context-aware model** that learns to convert between simplified and traditional Chinese characters under a denoising auto-encoder framework, **requiring no parallel data**.
- We propose to model the context information with pretrained language model and apply early exit mechanism when predicting the target character, which reduces the theoretical computation complexity, as well as improving the generalization ability of the model.
- We construct a test set with high quality text pairs. Extensive experiments testify that our model surpasses the existing publicly available simplified-traditional Chinese character conversion systems.

2. Approach

In this section, we introduce our **Unsupervised Adaptive Context-aware Conversion Model**, whose overall framework is shown in Figure 1. To **liberate the model from the need for parallel data**, we propose to train our model under a denoising auto-encoder framework, which comprises three components: the **Latent Generative Adversarial Encoder**, the **Context-aware Semantic Reconstruction Decoder**, and the **Token Prediction Module with Early Exit Mechanism**. The **encoder** looks up the original embedding of the characters and maps the embedded input into the latent semantic space, which should be as similar as possible to the counter-part character space achieved by a generative adversarial network (**GAN**). The **decoder** then restores the original semantic representation while considering contextual information. The **token prediction module** predicts the original input based on the contextual representation, which applies early exit mechanism to consider needs of different contextual levels. The parameters of the output layer within token prediction module are shared with the embedding layer of the encoder module, making the parameters more efficient.

The denoising auto-encoder training process is computed as follows,

1. Take the text of only traditional (denoted as I_t) Chinese characters as input and gets their embeddings (denoted as e_t) by looking up in the traditional Chinese character embedding table E_t.
2. Map the vector sequence e_t into the counterpart latent semantic space (simplified Chinese character space) with a linear transformation and yields e_s, which can be seen as the process of adding noise on the embedding level. A discriminator is introduced to make the transformed vectors indistinguishable from the real simplified Chinese character space under **GAN**.
3. Model the contextual information by feeding e_s into BERT pretrained on the simplified Chinese character text and gets the representation h_s. Note that the BERT is in accordance with the latent semantic space (e.g., simplified Chinese).
4. Restore the original traditional text sequence I_t by predicting the original text input based on h_s.

The detailed computation process is illustrated in Algorithm 1.

[1] https://ctext.org/

[2] The code and data is publicly available at https://github.com/NLPInBLCU/STCC

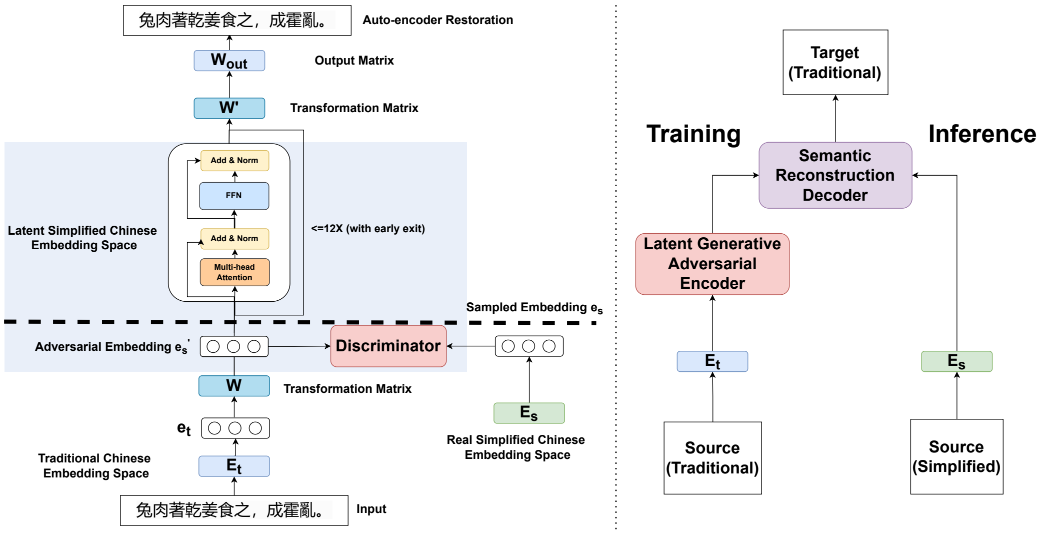

Figure 1: Overall Architecture of the proposed model taking simplified to traditional conversion as an example. On the left is the auto-encoder training procedure, on the right is the data flow of training and inference. The shadowed area indicates the latent semantic space of simplified Chinese. The part below the dashed line is the encoder. A discriminator is employed to ensure that the transformed embedding is mapped towards the simplified embedding space by the transformation matrix in **GAN**. The contextual information is modeled using a PLM, and the output representation is transformed back to the traditional Chinese character space.

2.1. Latent Generative Adversarial Encoder

We first separately learn the character embeddings of simplified and traditional Chinese character texts with unsupervised method word2vec (Mikolov et al., 2013) with monolingual data, yielding E_s and E_t for simplified and traditional Chinese.

Inspired by the approach of unsupervised machine translation (Lample et al., 2018), we propose to map the vectors in the traditional Chinese character space into the simplified Chinese character space (taking simplified to traditional conversion as an example) with a linear transformation $e'_s = W_t e_t$.

To learn more accurate transformation matrix, we apply the Generative Adversarial Network (GAN) (Goodfellow et al., 2014) framework following Lample et al. (2018), where a Discriminator is applied to judge whether the vector is from the target semantic space (E_s), the prediction score of which is calculated as,

$$s_D = \sigma(W_D \tilde{e}) \tag{1}$$

where W_D is the parameters of the discriminator, $\tilde{e}$ indicates the vector to be judged, $sigma$ is the sigmoid activation function that maps the output to the range within $0 \sim 1$. W_t plays the role of generator, whose objective is to confuse the discriminator from recognizing the transformed vectors e'_s. Following the canonical GAN training paradigm, the discriminator and the generator are trained iteratively, which stops at the point where the discriminator can hardly distinguish the difference between the generated vector and the real ones.

2.2. Context-aware Semantic Reconstruction Decoder

Although there is still a discrepancy between the transformed vectors e'_s and the real vectors from the simplified Chinese character space, the transformed vectors can be interpreted as distorted vectors with noise. By taking the surrounding context into consideration, our model can not only reduce the influence caused by the transformation process, but also make more accurate predictions when encountering one-to-many mappings. Therefore, we propose to use the pretrained language model trained on the target text (simplified Chinese character text) to model a more extensive range of context.

$$h_s = PLM(e'_s) \tag{2}$$

Concretely, we use roberta-classical-chinese [3] as the PLM in Equation 2 to obtain the semantic representation with context information.

After obtaining the contextualized representation h_s with the pretrained language model, we propose a symmetrical transformation process to map the vectors back to the original input semantic space one by one. Note that this transformation is applied

[3] https://huggingface.co/KoichiYasuoka/roberta-classical-chinese-base-char

Algorithm 1: Latent Denoising Auto-encoder Training Procedure

```
Require: Input Text $T$ in traditional Chinese character, Word Embedding Lookup Table for Traditional Chinese characters $E_t$ and Simplified Chinese characters $E_s$, Pretrained Language Model $PLM$, Transformation Matrix W and W', Discriminative Module $D$
Function LookupTransform($t$):
    $e_t \leftarrow E_t(t)$ // Look up the word embeddings of $t$ in $E_t$
    $e'_s \leftarrow We_t$ // Linearly transform $e_t$ to the simplified Chinese character semantic space
    return $e'_s$

for $i$ to number of discriminative training iterations do
    for each sample text $t$ in $T$ do
        $e'_s \leftarrow$ LookupTransform($t$)
        $e_s \leftarrow E_s(t)$ // Look up the word embeddings of $t$ in $E_s$ as $e_s$
        Update the $D$ model with the samples ($e_s$, 1) and ($e'_s$, 0) // (input, label)

for $i$ to number of generative training iterations do
    for each sample text $t$ in $T$ do
        $e'_s \leftarrow$ LookupTransform($t$)
        $l \leftarrow D(e'_s)$ // Predict the semantic space label of $e'_s$ with $D$
        Update the transformation matrix $W$ by minimizing $CrossEntropy(l, 1)$

for $i$ to number of training iterations do
    for each sample text $t$ in $T$ do
        $e'_s \leftarrow$ LookupTransform($t$)
        $h_s \leftarrow PLM(e'_s)$ // get context information with PLM
        $\tilde{h}_t \leftarrow W'_t h_s$ // transform back to the traditional Chinese space
        $p = softmax(W_{out}\tilde{h}_t)$
        Optimize the model parameters by minimizing $CrossEntropy(p, t)$
```

on the representation of each token, which is different from the setting of sequence-to-sequence generation. This transformed representation is then used to predict the original input characters, calculated as below:

$$\tilde{h}_t = W'_t h_s \tag{3}$$

$$p = softmax(W_{out}\tilde{h}_t) \tag{4}$$

where W_{out} indicates the prediction matrix. W'_t alongside W_{out} is applied so that we can take the transposed matrix of the Embedding lookup table E_t as W_{out} when sharing parameters.

Algorithm 2: Inference Procedure with Early Exit Mechanism

```
Require: Input Text $S$ in simplified Chinese character, Word Embedding Lookup Table for Simplified Chinese characters $E_s$, Pretrained Language Model $PLM$, Transformation Matrix W and W', Exit threshold $threshold$
for each sample text $s$ in $S$ do
    $e_s \leftarrow E_s(s)$
    // lookup the embeddings of $s$ in $E_s$
    $h_0 \leftarrow e_s$
    for $j$ in 1 to total layer number do
        $h_j \leftarrow PLM_j(h_{j-1})$ // get the $j$-th layer hidden states with $PLM_j$
        $\tilde{h}_j \leftarrow W_j h_j$ // transform to the traditional Chinese space
        $p_j = softmax(W_{out}\tilde{h}_j)$
        if $max(p_j) > threshold$// exit when exceeding threshold
        then
            return $p_j$
```

After getting the prediction probability p, we can calculate the cross entropy between p and input I_t as the loss function,

$$loss = -\Sigma_i I_t^i logp \tag{5}$$

where I_t^i indicates the i-th input token, which also serves as the restoration target in our auto-encoder.

2.3. Inference with Early Exit Mechanism

During inference, only the decoder part of the model is needed. The traditional to simplified decoder is taken from the model trained on only simplified Chinese documents, while the simplified to traditional decoder is taken from the model trained on only traditional Chinese document.

However, the aforementioned model suffers from the over-computation problem during inference due to the fact that many characters remain unchanged after conversion. This means that shallow context information is sufficient for such characters, and forwarding all the hidden layers of the pretrained language models is unnecessary. To address this issue, we propose to introduce early exit mechanism to our model. Inspired by DeeBERT (Xin et al., 2020), we add an additional transformation module to each candidate layer that transforms the hidden states of that layer to the space of the last hidden layer, which is then used for prediction, calculated as follows:

$$\tilde{h}_j = W^j_{exit} h_j \tag{6}$$

where j indicates the j-th layer, W_{exit}^{j} indicates the transformation matrix for the j-th layer.

During training, all the candidate layers will predict the original character token based on their transformed hidden states $\tilde{h}_j$,

$$\tilde{p}_j = softmax(W_{out}\tilde{h}_j) \tag{7}$$

Note that the parameters for output W_{out} is shared across different layers including the final layer in Equation 4. To optimize the transformation parameters of each layer, the new loss function becomes the sum of the cross entropy for each layer,

$$loss_{exit} = -\Sigma_i \Sigma_j I_t log \tilde{p}_j \tag{8}$$

During Inference, each candidate layer in the decoder module predicts the traditional Chinese character conversion step by step, and the prediction process stops when the prediction score (confidence) of a layer achieves a certain threshold, which is a pre-defined empirical hyper-parameter. This early exit mechanism can not only reduce the theoretical computation complexity, but also help alleviate the over-fitting problem. The detailed computation process is shown in Algorithm 2.

Additionally, to cope with the out-of-vocabulary (OOV) problem, we propose a technique called “OOV Skipping”, which involves using the original input as the conversion result for characters that appear less than a predefined threshold (e.g., 5 times). This is based on the observation that the Chinese character simplification process mostly focus on the frequently used characters, which means it is reasonable to keep infrequent characters the same before and after conversion.

3. Experiment

In this section, we present the experimental details and extensive analysis of the results.

3.1. Data

We collect the data from an online open-access digital library CTEXT.[4] We utilize the “zhon”[5] package for sentence segmentation. The simplified to traditional Chinese conversion model is taken from the Inference module of the model trained on traditional Chinese documents of 1,141,001 sentences. The traditional to simplified Chinese conversion model is taken from the Inference module of the model trained on simplified Chinese documents of 1,141,001 sentences. To highlight the one-to-many problem, we choose 20,000 high-quality sentence pairs with one-to-many characters as the test set that are proof-read by human.

3.2. Setting

We use gensim word2vec [6] to generate word embeddings separately for traditional and simplified Chinese characters, the dimensions of which are 768 and 1024 separately for base and large model. The window size is 5. The minimum word frequency is 5, resulting in a vocabulary size of 25,621 and 29,513 for simplified and traditional Chinese. The number of training epochs is 5. The implementation of the Latent Generative Adversarial Encoder follows MUSE (Lample et al., 2018). The number of training epochs and refinement rounds are both 5. For the discriminator, the layer number is 2 with a hidden dimension of 2048. SGD optimizer is used to train the Latent Generative Adversarial Encoder, and the learning rate is set to 0.1 with a batch size of 32. For the whole auto-encoder training stage, the maximum sentence length is set to 128 and the batch size is set to 8. The number of epochs for training is 1. We use AdamW optimizer (Kingma and Ba, 2015), and set the learning rate to 2e-5. The threshold value for early exit is set to 0.996.

3.3. Baseline

In this subsection, we introduce the baseline models that we use for comparison. Only publicly available baselines were considered. The baselines consist of four parts, software, websites, unsupervised machine translation models and ChatGPT.

- Software: we apply Open Chinese Convert (**OpenCC**)[7], Microsoft Word- (**MS Word**), **zhconv**[8] and **Pylangtools**[9] as the publicly available baseline software to compare with.
- Website: we apply **JianFan**[10], **AIES**[11], **KJSON**[12] as the websites providing simplified and traditional Chinese character conversion as baselines to compare with.
- Unsupervised machine translation: We apply four word embedding based unsupervised or semi-supervised machine translation methods **SVD** (Schönemann, 1966), **MUSE** (Lample et al., 2018), Artetxe et al. (2018b) and Artetxe et al. (2018a). Artetxe et al. (2018b) has both unsupervised and semi-supervised settings.
- ChatGPT[13] (gpt-3.5-turbo): we use the prompt in English which is “Translate the following sen-

[4] https://ctext.org

[5] https://github.com/tsroten/zhon

[6] https://radimrehurek.com/gensim/models/word2vec.html

[7] https://github.com/BYVoid/OpenCC

[8] https://github.com/gumblex/zhconv

[9] https://pypi.org/project/pylangtools/

[10] jianfan.hwxnet.com

[11] aies.cn

[12] www.kjson.com/office/zhcn_zhtw

[13] http://chat.openai.com/

Method	S-to-T	T-to-S
OpenCC	95.64	97.56
MS Word	96.89	97.51
zhconv	97.96	99.49
pylangtools	96.77	97.79
Website		
Jianfan	94.26	94.98
AIES	97.72	98.84
KJSON	95.72	97.24
Unsupervised MT		
SVD	89.61	89.82
MUSE (Lample et al., 2018)	93.91	93.93
Artetxe et al. (2018a)	93.74	93.88
Artetxe et al. (2018b)(unsup)	94.22	94.21
Artetxe et al. (2018b)(semi)	94.25	94.28
ChatGPT	79.22	79.21
Proposal(base)	**98.45**	**99.51**
Proposal(large)	**98.57**	**99.61**

Table 1: Character-wise Accuracy Comparison of Unsupervised Baselines and Proposed Method for Chinese Character Conversion. We apply character-wise accuracy because this is a strictly one-to-one conversion task, which is different from sequence-to-sequence translation. The conversion directions are indicated by "S-to-T" for converting from simplified to traditional Chinese characters and "T-to-S" for the reverse direction. Our proposed method achieves the highest accuracy compared to the baselines for both directions.

tence into traditional Chinese” for simplified to traditional Chinese conversion.

3.4. Result

In this subsection, we present the character-wise accuracy of our proposed model compared with the baseline models in Table 1. It shows that our model achieves the highest accuracy for both directions of Chinese character conversion. Remarkably, even the proposed base model outperforms all the baselines, providing compelling evidence for the effectiveness of our proposed method. Moreover, even with initially aligned seed tokens, Artetxe et al. (2018b) reaches a bottleneck because of the lack of considering context information. Furthermore, we notice that although large language models like ChatGPT can perform simplified-traditional Chinese conversion, it does not yield satisfactory results under zero-shot setting.

Additionally, we note that converting from simplified to traditional Chinese characters is more challenging than the opposite direction due to the fact that one simplified Chinese character can have multiple traditional Chinese character counterparts, whereas one traditional Chinese character usually maps to only one simplified Chinese character. As a result, converting from simplified Chinese characters requires disambiguation, making it more complex.

3.5. Ablation Study

In this part, we show the results of each designed module in our model. The basic module named Latent Generative Adversarial Encoder (mentioned in section 2.1) learns a linear mapping from the simplified Chinese character space to the traditional Chinese character space with GAN. From the result we can see that because this kind of static model does not consider the target context, it does not yield very satisfactory accuracy (93.91), which is just slightly better than the performance of OpenCC, which also applies a static mapping strategy.

By further applying our proposed context-aware semantic reconstruction decoder (mentioned in section 2.2), the accuracy improves significantly reaching 96.79, which is comparable to the strong baselines (e.g., *MS Word* 96.89, *pylangtools* 96.77, etc.). This improvement validates the effectiveness of our proposed denoising auto-encoder training method, which restores the original input by considering semantic context to alleviate the noise introduced by linear embedding space transformation. Additionally, we observe that sharing parameters between the encoder and the decoder by unifying the restored embedding space and the real input embedding space further improves the accuracy to 97.13, surpassing most of the baselines except for the website `aies.cn`.

Furthermore, we can observe that applying early exit not only reduces the theoretical computation complexity, but also improves the conversion accuracy to 98.35, which is higher than all the baselines. We argue that this is because that dynamically selecting the exiting point can improve the generalization ability, which is especially important for those infrequent characters. Finally, the post processing that directly predicts the infrequent characters to themselves gives a small improvement. We assume that this is because although this operation can fix some errors made by the model, the frequency of occurrence of such characters in the corpus is relatively small.

3.6. Analysis

3.6.1. Multiple Mapping

In this part, we analyse the multiple mapping phenomenon in simplified-to-traditional Chinese character conversion task. We separately show the accuracy for different mapping patterns in Figure 2 with bar plot, while their corresponding sentence appearance ratio is shown with line plot. The ratio is calculated as the number of sentences where the

Module	S-to-T	T-to-S
Latent Generative Adversarial Encoder	93.91	93.93
+ Context-aware Semantic Reconstruction Decoder	96.79	97.66
+ Encoder-Decoder Parameter Sharing	97.13	98.01
+ Early Exit	98.45	99.52
+ OOV Skipping	98.57	99.61

Table 2: This table displays the results of an ablation study where modules were added one by one to evaluate their impact on the performance of the proposed model.

pattern appears divided by the total number of sentences, therefore they do not sum to 1, as different patterns can appear within the same sentence.

From the results we can see that generally the more complex the mapping pattern is, the lower the accuracy gets, which is in accordance with the intuition. However, the gap among 1 vs. 2, 1 vs. 3 and 1 vs. 4 is quite small, which are less than 0.02 percent. The most complicated mapping pattern (m vs. n) would influence the accuracy the most. Luckily, generally the more complex the pattern is, the lower ratio it takes in the whole dataset. The m vs . n pattern only takes a small part of the data, which means it would not severely influence the overall performance.

3.6.2. Backbone PLM

In this part, we show the results obtained by testing our model with different pretrained language models as backbone in Figure 3. Specifically, we evaluated guwen-bert [14], roberta-classical-chinese and bert-ancient-chinese (Wang and Ren, 2022), where both base model and large model of guwen-bert and roberta-classical-chinese are applied, denoted respectively as “base” and “large”. From the results we can see that roberta based model is generally better than bert based model. More importantly, large models generally gives better result over base models, which is expected. However, the gap between base model and large model is not very significant, indicating that base models could also be used when faster inference speed is a more important factor to consider.

3.6.3. Sentence Length

In this part, we investigate the effect of maximum sentence length. We evaluate the performance of the proposed model with increasing maximum sentence length and compare it with the baseline encoder-only model. The results are shown in Figure 4. The results indicate that as the maximum sentence length increases, the accuracy of the proposed model first improves and then stabilizes after a certain length. This suggests that longer context can provide more information for predicting the target character, which is in accordance with our hypothesis. However, the accuracy of the baseline encoder-only model remains the same across different maximum sentence lengths, as it does not consider context. Overall, these results demonstrate the importance of context in the simplified-to-traditional Chinese character conversion, and highlight the effectiveness of the proposed model in utilizing context to improve performance.

3.7. Case Study

In this subsection, we provide some concrete examples for simplified Chinese character to traditional Chinese character conversion and compare the results of our proposed method with two strong baselines. The correct predictions are in black, while the wrong predictions are in red.

In the first set of examples, our model gives the correct predictions in the second two cases, while gives wrong predictions in the other two cases. We assume that this is because the usage of last three traditional Chinese character regarding to the character “里” (inner) is very similar, which all follow the appearance of “表”(surface), forming the word “表里” still in usage nowadays. In fact, the second and third case “裡” and “裏” in traditional Chinese character systems are variant Chinese characters, expressing the same meaning with different forms, while the fourth case “里” is actually misused because such occurrence usually means the residence place or distance unit rather than the normally referred meaning of inner. For the first case, our model correctly predicts “里”, which means the distance unit. However, the strong baseline models (zhconv and aies.com) give the wrong characters (“裡” and “裏”). This shows the superiority of our model on contextual semantic modeling.

In the second set of examples, the simplified Chinese character “斗” has two distinct meanings, weighting bucket “斗” (extended in meaning the Plough star) pronounced dou3 and fighting “鬥” pronounced dou4. Our model gives the correct predictions in both of the cases, while the strong baselines incorrectly predict the character for fighting “鬥” in the first case.

The third set of examples involves the simplified Chinese character “饥”, which has two closely re-

[14] https://github.com/Ethan-yt/guwenbert

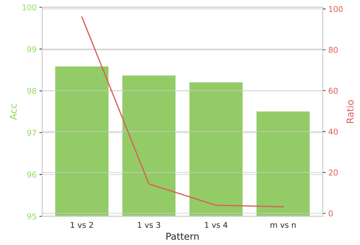

Figure 2: Accuracy and Appearance Ratio of Mapping Patterns in Simplified-to-Traditional Chinese Character Conversion.

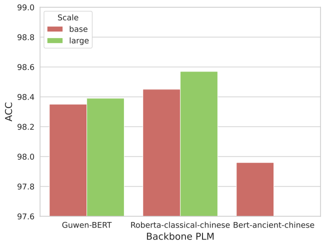

Figure 3: Accuracy of different backbone PLMs for simplified to traditional Chinese conversion. Both large and base models are included, except for "Bert-ancient-Chinese".

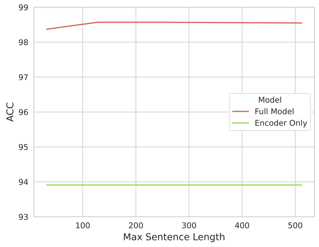

Figure 4: Accuracy of Proposed Model and Encoder-Only Baseline for Different Maximum Sentence Lengths in Simplified-to-Traditional Chinese Character Conversion.

Simplified	Traditional	Proposal	zhconv	aies.cn
故国有贤君，折冲万**里**。	故國有賢君，折沖萬**里**。	里	裡	裏
中外若一，事无表**里**。	中外若一，事無表**裏**。	裏	裡	裏
文貌情用，相为内外表**里**。	文貌情用，相為内外表**裡**。	裏	裡	裏
表**里**相资，古今一也。	表**里**相資，古今一也。	裏	裡	裏
星莫大于大辰，北**斗**常星。	星莫大於大辰，北**斗**常星。	斗	鬥	鬥
朝有变色之言，则下有争**斗**之患。	朝有變色之言，則下有爭**鬥**之患。	鬥	鬥	鬥
百官**饥**饿，河内太守张杨使数千人负米贡饷。	百官**飢**餓，河內太守張楊使數千人負米貢餉。	飢	飢	饑
四时不出，天下大**饥**。	四時不出，天下大**饑**。	饑	飢	饑

Figure 5: Examples of Simplified-to-Traditional Chinese Character Conversion with Incorrect Predictions Highlighted in Red. The comparison targets are in bold. The translations are provided in the Appendix.

lated meanings: hungry "飢" and poor harvest "饑". These meanings are difficult to distinguish, but our proposed model correctly predicts the appropriate character for each meaning. In contrast, both baselines predict the same character for both meanings, indicating a failure to differentiate between the two.

4. Related Work

Although simplified-traditional Chinese character conversion is a practical and important task, there has not been much research on it. Early works on simplified-traditional Chinese character conversion apply table lookup methods (Li et al., 2010). This kind of methods consider little context information, and thus yielding weak performance. Hao and Zhu (2011) also address the importance of the one-to-many cases in simplified to traditional Chinese character conversion. However, their proposed Fused Conversion Algorithm from Multi-Data resources only considers n-gram statistical model, which is too shallow to model enough contextual semantics. Similar to their work, Chen et al. (2011) and Shi et al. (2011) propose to use log-linear model that takes features such as language models and lexical semantic consistency weighs. Xu et al. (2017) still apply statistical conversion model, but provides a proof reading web interface.

5. Conclusion

In this work, we propose an unsupervised adaptive context-aware model for the simplified-traditional Chinese character conversion task, which not only liberates the model from the needs of parallel data, but also endows it with better adaptation ability. To alleviate the one-to-many problem, we propose to introduce PLM for contextual semantic modeling in a reconstruction decoder, which restores the original input in a denoising auto-encoder framework. Based on the observation that different characters may require different levels of semantic modeling, we propose to apply early exit mechanism for inference, which can not only reduce theoretical computation complexity, but also improve the generalization ability. Experiments on the constructed simplified-traditional Chinese character conversion test set show the superiority of our proposed model against strong baselines. Extensive analysis testifies that modeling contextual information with pretrained language model can indeed help distinguish different meanings in the one-to-many scenarios.

Acknowledgements

This research project is supported by the National Natural Science Foundation of China (No. 62306045), Science Foundation of Beijing Language and Culture University (supported by "the Fundamental Research Funds for the Central Universities") (No. 21YBB19)

6. Bibliographical References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5012–5019.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798.

Yidong Chen, Xiaodong Shi, and Changle Zhou. 2011. A simplified-traditional chinese character conversion model based on log-linear models. In *2011 International Conference on Asian Language Processing*, pages 3–6.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks.

Tianyong Hao and Chunshen Zhu. 2011. Simplified-traditional chinese character conversion based on multi-data resources: Towards a fused conversion algorithm. volume 3, pages 50 – 56.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Min-Hsiang Li, Shih-Hung Wu, Ping-che Yang, and Tsun Ku. 2010. (Chinese characters conversion system based on lookup table and language model) [in Chinese]. In *Proceedings of the 22nd Conference on Computational Linguistics and Speech Processing (ROCLING 2010)*, pages 113–127, Nantou, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Peter H Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Xiaodong Shi, Yidong Chen, and Xiuping Huang. 2011. Key problems in conversion from simplified to traditional chinese characters. In *International Conference on Asian Language Processing*.

Pengyu Wang and Zhichen Ren. 2022. The uncertainty-based retrieval framework for ancient chinese cws and pos. In *Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages*, pages 164–168.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2246–2251, Online. Association for Computational Linguistics.

Jiarui Xu, Xuezhe Ma, Chen-Tse Tsai, and Eduard Hovy. 2017. STCP: Simplified-traditional Chinese conversion and proofreading. In *Proceedings of the IJCNLP 2017, System Demonstrations*, pages 61–64, Tapei, Taiwan. Association for Computational Linguistics.