Istic Neural Machine Translation System for EvaHan 2023

Ningyuan Deng dengny2022@istic.ac.cn
Shuao Guo guosa2021@istic.ac.cn
Yanqing He* heyq@istic.ac.cn
Research Center of Information Theory and Methodology, Institute of Scientific and Technical Information of China, Beijing 100038, China

Abstract

This paper presents the system architecture and the technique details adopted by Institute of Scientific and Technical Information of China (ISTIC) in the evaluation of First Conference on EvaHan(2023). In this evaluation, ISTIC participated in two tasks of Ancient Chinese Machine Translation: Ancient Chinese to Modern Chinese and Ancient Chinese to English. The paper mainly elaborates the model framework and data processing methods adopted in ISTIC's system. Finally a comparison and analysis of different machine translation systems are also given.

1. Introduction

This paper presents a detailed overview of the machine translation system of the Institute of Scientific and Technical Information of China (ISTIC) in the EvaHan (2023) evaluation task. ISTIC participated in the Ancient-Modern Chinese and Ancient-English translation tasks.In this evaluation Google Transformer[1] is used as the baseline system. Open source monolingual data is forward translated to construct a pseudo-parallel corpus to expand the training set released by EvaHan (2023) Evaluation side. Data pre-processing includes special character filtering, sentence de-duplication, length-ratio filtering and Pinyin coding. In the construction of the system model, a context-awareness-based approach[2] encode the context as an additional neural network. Then the model integration method are used to obtain the final translation results.

The structure of this paper is structured as follows: Section 2 introduces the system framework and technical approach adopted by ISTIC. Section 3 presents the experimental setting and results. Finally we conclude our work in Section 4.

[1] Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30 (NIPS 2017), pp. 5998 – 6008 (2017)

[2] Fernandes, P., Yin, K., Neubig, G., & Martins, A. F. T. (2018). Measuring and Increasing Context Usage in Context-Aware Machine Translation. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (pp. 1114-1123).

Proceedings of ALT2023: First Workshop on Ancient Language Translation, pages 34–42
September 5, 2023, Macau SAR, China.

2. System

ISTIC participated in Ancient-Modern Chinese task (a2m) and Ancient-English task (a2e). Figure 1 shows the system architecture of our machine translation including data augmentation, data preprocessing, data set partition, model training, model inference and data post-processing.

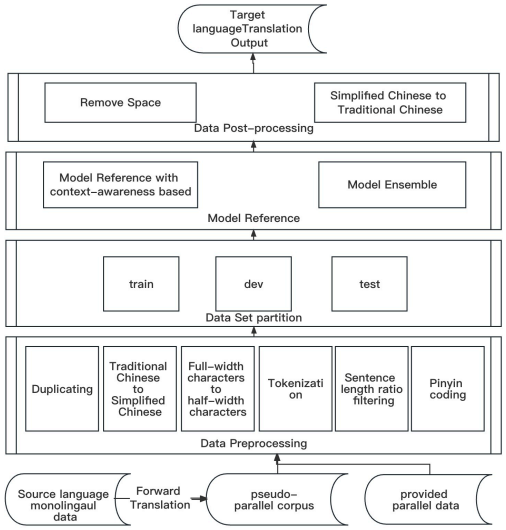

Figure 1:flow chart of our machine translation system

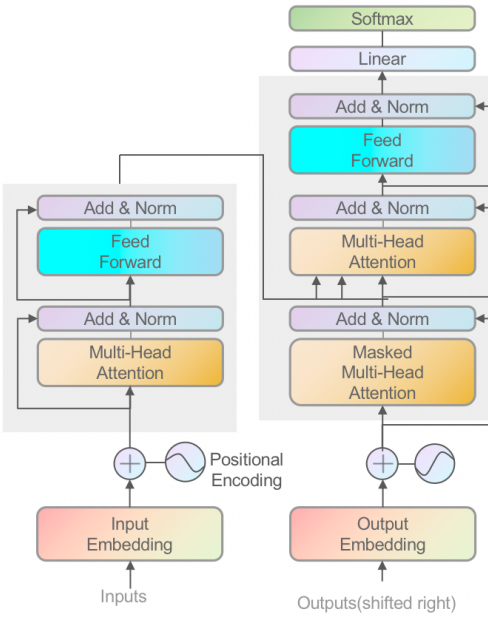

Figure 2:Transformer[2] model structure

2.1. Data augmentation

Forward translation is one of the common ways of data augmentation[3].We first train the translation models of a2m and a2e using the released data. Then ancient Chinese monolingual data is collected from internet and translated by the above two machine translation models to construct pseudo-bilingual pairs. Finally the released parallel sentence pairs are merged with pseudo-bilingual pairs as the final data.

2.2. Data Preprocessing

The main stages of preprocessing are as follows.
1. Duplicating:We remove repetitive sentences to reduce the training time of machine translation models
2. Traditional Chinese to Simplified Chinese: By converting traditional Chinese to simplified Chinese we can obtain a uniformly encoding for each same Chinese word.

[3] Nishant Kambhatla, Logan Born, and Anoop Sarkar. 2022. CipherDAug: Ciphertext based Data Augmentation for Neural Machine Translation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 201 – 218, Dublin, Ireland. Association for Computational Linguistics.

3. Full-width characters to half-width characters: Because full-width characters and half-width characters have different Unicode encoding, we converts full-width characters to half-width characters in order to avoid inconsistent rendering of text, or even garbled code.
4.Tokenization: We tokenize the data into meaningful units, so that the translation system can understand and process the input text more accurately and make the vocabulary smaller.
5. Sentence length ratio filtering: The sentence lengths ratio of the source language and target language can help to filter poor bilingual pairs.
6. Pinyin coding: Chinese characters may correspond to multiple pronunciations.So we converts Chinese characters into corresponding Pinyin coding in order to more accurately match the similarities and correlations between the source and target language.

2.3. Data Set Partition

We split all the preprocessed data into: a training set, a test set, and a validation set.The training set and the validation set are used to train the supervised machine translation model and adjust the parameters.The test set is employed to verify whether the trained model has the same effect in other data. The partition ratio of the dataset is train:test:dev=90:5:5.

2.4. Model Inference

Our system is based on Google Transformer[4]. As shown in Figure 2, Transformer comprises two components: an encoder layer (with self-attention and fully connected layers) and a decoder layer (with self-attention, encoder-decoder attention, and fully connected layers), and each of them consists of 6 modules. The model adopts the self-attention mechanism and realizes algorithm parallelism to improve translation quality.

The data of the evaluation tasks are all sentences pairs and lack contexts information. Therefore a context-awareness-based approach is employed and multi-encoder concatenate[5] the source sentence and their contexts, such as inside-context, outside-context and Gaussian noise context.

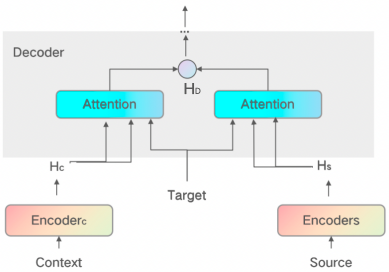

Figure 3. Inside-context method.

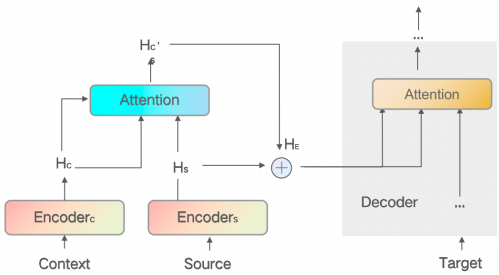

Figure 4 Outside-context method.

(1)Inside-context: We use Pinyin coding of source language or source language itself as its context.In a2m tasks,And Since ancient Chinese has kanji, homophones, supplemental Pinyin can pass kanji information .In a2e task ,Chinese characters does not make full use of the internal semantics of Chinese, while the Latin alphabet has prefix suffixes such as de and an. Therefore,

[4] Facebook Research. Fairseq: A Fast, Extensible Toolkit for Sequence Modeling. Github, 2016, https://github.com/facebookresearch/fairseq.
[5] Bei Li, Hui Liu, Ziyang Wang, Yufan Jiang, Tong Xiao, Jingbo Zhu, Tongran Liu, and Changliang Li. 2020. Does Multi-Encoder Help? A Case Study on Context-Aware Neural Machine Translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3512 – 3518, Online. Association for Computational Linguistics.

supplementing the Pinyin coding can make the correspondence with the Latin alphabet, so as to show more of the internal characteristics of Chinese characters, and establish a relationship with the English, so as to solve the problem of learning bottlenect and parameter bottlenect. As shown in Figure 3, firstly, the decoder can attend to two encoders respectively ,which are H_s (the hidden layer of source sentence)and H_c (the hidden state of the contexts).Then in decoder layer,we concat H_c and H_s with H_t (the hidden layer of target sentence) to form $H_{c'}$ and $H_{s'}$ .Finally the gating mechanism inside the decoder is employed to obtain H_D (the fusion vector). In the Inside approach, Target is the query, Hs and Hc represent key/value.

$$H_{c'} = Concat(H_c, H_t)$$

$$H_{s'} = Concat(H_s, H_t)$$

$$H_D = MultiHead(H_{c'}, H_{s'})$$

(2)Outside-context: We also use Pinyin coding of source language or source language as its context.As shown in Figure 4, firstly we convert current source sentence and its context into new vectors H_s (the hidden layer of source sentence)and H_c (the hidden state of the contexts) . Through the attention mechanism of the encoder,we concat H_c and H_s to form a new vector ,called $H_{c'}$(the hidden state of the attention part of the encoder). Then the attention output ($H_{c'}$) and the source sentence(H_s) are fused by a gated sum to form H_E(the mutihead encoder layer).**. In the Outside approach, Hs is the query and Hc is thekey/value.**

$$H_{c'} = Concat(H_c, H_s)$$

$$H_E = MutiHead(H_{c'} \quad H_s)$$

(3)Gaussian-noise-context:It is similar to outside-context method.It add Gaussian noise to the encoder output and combining the context with Gaussian noise.

2.5. Model ensemble

Model ensemble[6] can improve the generalization ability of the final model by fusing multiple trained models. Then ultimate result involves weighted average of probability distribution for predictions, which combines the learning capabilities of all the individual models

2.6. Data post-processing

In the Ancient Chinese to Modern Chinese task the first step of data post-processing is to remove space and the second step is to restore simplified Chinese characters to traditional Chinese characters to satisfy the submission requirements. In the Ancient Chinese to English task the first step of data post-processing is to remove extra space.Secondly we restore the case of the English results.

3. Experiments

The aims of experiment are to verify (1) whether context awareness models can provide more information gain; (2) which one of the inside-context and outside-context model performs

[6] Ganaie M A, Hu M, Malik A K, et al. Ensemble deep learning: A review[J]. Engineering Applications of Artificial Intelligence, 2022, 115: 105151.

better; (3) which one of the source laguage context and Pinyin coding contexts performs better. We use BLEU4[7] to evaluate the quality of the translations, which is automatic evaluation index of machine translation commonly used now.

3.1. System Settings

We trained our machine translation model by the Fairseq sequence modeling toolkit of PyTorch. The main parameters are set as follows: each model uses 1-3 GPUs, batch size is 2048, parameter update frequency is 1, learning rate is 5e-4, and the number of warmup steps is 4000. Maximum number of tokens is 4096. Self-attention mechanism uses 16 heads.The dropout is 0.3. BPE is 32K.Loss function is label smoothed cross entropy.Adam betas is (0.9,0.997). Maximum epoch is 40. Initial learning rate is 0.0005,Context-aware learning rate is 0.0001.

3.2. Data Preprocessing

The data include the released parallel data and external data of monolingual languages. For the Ancient-Chinese-to-English machine translation task we use the released data and Twenty-four Histories (ancient Chinese monolingual data). For the Ancient-Chinese-to-English machine translation task, we adopte the released data and Zizhi Tongjian (ancient Chinese monolingual data). Forward translation generates the pseudo-parallel[8] corpus as supplementary data. Both forward translation and the released data are preprocessed to reduce data noise:
1. Duplicating;
2. Traditional Chinese to Simplified Chinese: zhconv[9] is used to convert;
3. Full-width characters to half-width characters: Niutrans[10] preprocessing toolkit;
4.Tokenization: urheen[11] for modern Chinese and jiayan[12] package for ancient Chinese;
5. Sentence length ratio filtering: we retain sentences with length ratio in [0.1,10];
6. Pinyin coding: xpinyin[13] is used to generate Pinyin for source sentences;

The number of sentences with preprocessing results is listed in Table 1. The partition of sentence is shown in Table 2.

Table 1. The statistics of preprocessed data.

Type	Before preprocessing	After preprocessing

[7] Papineni K, Roukos S, Ward T, et al. Bleu: a method for automatic evaluation of machine translation[C]//Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002: 311-318.
[8] Haddow B, Bawden R, Barone A V M, et al. Survey of low-resource machine translation[J]. Computational Linguistics, 2022, 48(3): 673-732
[9] nobodxbodon.zhconverter.https://github.com/nobodxbodon/zhconverter
[10] NiuTrans. NiuTrans: An Open Source Neural Machine Translation Toolkit. https://github.com/NiuTrans/NiuTrans.
[11] Chinese Academy of Sciences, Institute of Automation. Chinese Information Processing Software Download. https://www.nlpr.ia.ac.cn/cip/software.html.
[12] Jiayan. Jiayan: Chinese word segmentation in Python.https://github.com/jiaeyan/Jiayan.
[13] lxneng. Xpinyin: Convert chinese hanzi to pinyin.https://github.com/lxneng/xpinyin.

Table 2. The partition of data

	（Sentence pair）	（Sentence pair）
Released bilingual data in a2m	307,494	303,164
Zizhi Tongjian data	319,883	312,389
Released bilingual data in a2e	5,899	5,898
Twenty-four Histories data	305,163	305,162

Task	train	dev	test
a2m	553,998	30,777	30,777
a2e	279,954	15,553	15,53

3.3. Baseline systems

We use three Transformer model architectures as baseline systems in this evaluation:
(1) Transformer: It is based on the Transformer architecture, which is suitable for handling medium-sized translation tasks. Compared with the larger model, it requires less training data and computational resources, but may be inferior in performance.
(2) Transformer_wmt_en_de_big: It is suitable for multilingual translation tasks.
(3) Transformer_wmt_en_de_big_t2t: It is an end-to-end machine learning framework with the Tensor2Tensor (T2T) framework for training.

3.4. Experimental Results

3.4.1. Context Performance

Tables 3-4 show the results of baseline model and context-awareness-based model for Ancient Chinese to Modern Chinese task and Ancient Chinese to English task. In baseline model 1-3, we only use the released bilingual corpus while in baseline model 4 we use both released corpus and pseudo-bilingual data.The difference between baseline models is model architectures. Baseline model 1-3 are transformer, transformer_wmt_en_de_big, and transformer_wmt_en_de_big_t2t. And the baseline model 4 use the same architectures as baseline model 3. Inside-context system means the inside-context awareness model. Outside-context system represents the outside context-awareness model, Gaussian context means gaussian noise as context. Src means source language sentence as context, Src.Pinyin coding means the Pinyin coding of source language as context.

Table 3. Performance comparison of a2m
(Baseline1-3 only use released data, other models use whole data)

System	BLEU (%)
Baseline 1(arch = transformer)	37.35
Baseline 2(arch = transformer_wmt_en_de_big)	37.80
Baseline 3(arch = transformer_wmt_en_de_big_t2t)	38.03
Baseline 4(transformer_wmt_en_de_big_t2t+ pseudo-bilingual)	38.15
Transformer(Src)	38.57
Inside-context(Src)	38.97
Outisde-context(Src)	39.26
Transformer(Src.Pinyin coding)	37.81

Inside-context(Src.Pinyin coding)	38.98
Outisde-context(Src.Pinyin coding)	38.92
Gaussian-context(Src.Pinyin coding)	39.00

From Table 3 the most effective model among baseline 1-4, is the baseline 4 which is based on Transformer_wmt_en_de_big_t2t plus pseudo-bilingual data and achieves a BLEU score of 38.15, improving up to 0.8% BLEU compare with baseline 1. By comparing the Transformer(Src) with Basline 4 ,we found just generating bpe dict with context information also helps a little though the constructure of modesl is not changed. Among context-awareness model of source language as contexts, the outside-context performs the better with a BLEU score of 39.26. When Pinyin coding is used as contexts, the inside-context performs best with a BLEU score of 38.98.

Table 4. Performance comparison of a2e
(Baseline1-3 only use released data, other models use whole data)

System	BLEU (%)
Baseline 1(arch = transformer)	4.19
Baseline 2(arch = transformer_wmt_en_de_big)	5.32
Baseline 3(arch = transformer_wmt_en_de_big_t2t)	6.09
Baseline 4(arch=transformer_wmt_en_de_big_t2t+ pseudo-bilingual)	17.44
Transformer(Src)	18.16
Inside-context(Src)	18.38
Outisde-context(Src)	18.18
Transformer(Src.Pinyin coding)	17.49
Inside-context(Src.Pinyin coding)	18.48
Outisde-context(Src.Pinyin coding)	18.46
Gaussian-context(Src.Pinyin coding)	18.51

From Table 4 data enhancement also effectively improves translation performance. Transformer_wmt_en_de_big_t2t + pseudo-bilingual data achieves a BLEU score of 17.44, improving up to 13.25% BLEU compare with baseline 1. And in both type of contexts, inside-context preforms better But the Gaussian performs best.

3.4.2. Ensemble Performance

Table 5-6 compare the result of context-awareness-based model with ensemble. The strategy is seperately combined with inside-context model and gaussian model. The ensemble approach did not perform well both in a2m and a2e. After the ensemble our model BLEU score dropped by 0.3%.

Table 5. Comparison of context-awareness model for a2m

System	BLEU （Src）	BLEU (Src.Pinyin coding）
Inside-context+ensemble	38.58	38.62

Gaussian-context+ensem-ble	38.85

Table 6. Comparison of context-awareness model for a2e

System	BLE（Src）	BLEU（Src.Pinyin coding）
Inside-context+ensemble	18.14	18.12
Gaussian-context+ensemble	18.51	

We submitted the result of Gaussian-context+ensemble for Ancient Chinese to Modern Chinese task and the result of Gaussian-context+ensemble for Ancient Chinese to English tasks.

4. Conclusion

This paper presents the main methods of ISTIC's Machine Translation System in Eva-Han(2023). Our model is based on the Transformer architecture and context-awareness model. We used Forward translation to enhance the training data. This strategy works well when data is scarce such as in a2e task, and has a little boost when data is more sufficient, such as in a2m task.The context-awareness methods can effectively improve the translation performance whether the contexts are source language or source language's Pinyin coding. But the ensemble did not work very well.

For future work, there are many interesting directions. Firstly we will study how to mine the linguistic knowledge between ancient Chinese and modern Chinese and integrate it into the context informtion. Secondly we will continue to improve the contexts-awareness based model both on encoder layer and decoder layer.

References

Bei Li, Hui Liu, Ziyang Wang, Yufan Jiang, Tong Xiao, Jingbo Zhu, Tongran Liu, and Chang-liang Li. 2020. Does Multi-Encoder Help? A Case Study on Context-Aware Neural Machine Translation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3512–3518, Online. Association for Computational Linguistics.

Chinese Academy of Sciences, Institute of Automation. Chinese Information Processing Software Download. https://www.nlpr.ia.ac.cn/cip/software.html.

Facebook Research. Fairseq: A Fast, Extensible Toolkit for Sequence Modeling. Github, 2016, https://github.com/facebookresearch/fairseq.

Fernandes, P., Yin, K., Neubig, G., & Martins, A. F. T. (2018). Measuring and Increasing Con-text Usage in Context-Aware Machine Translation. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Huma Language Technologies, Volume 1 (Long Papers) (pp. 1114-1123).

Ganaie M A, Hu M, Malik A K, et al. Ensemble deep learning: A review[J]. Engineering Appl ications of Artificial Intelligence, 2022, 115: 105151.

Haddow B, Bawden R, Barone A V M, et al. Survey of low-resource machine translation[J]. C omputational Linguistics, 2022, 48(3): 673-732

lxneng. Xpinyin: Convert chinese hanzi to pinyin.https://github.com/lxneng/xpinyin.

Jiayan. Jiayan: Chinese word segmentation in Python.https://github.com/jiaeyan/Jiayan.

Nishant Kambhatla, Logan Born, and Anoop Sarkar. 2022. CipherDAug: Ciphertext based Data Augmentation for Neural Machine Translation. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 201–21 8, Dublin, Ireland. Association for Computational Linguistics.

NiuTrans. NiuTrans: An Open Source Neural Machine Translation Toolkit.https://github.com/Niu Trans/NiuTrans.

Nobodxbodon.zhconverter.https://github.com/nobodxbodon/zhconverter

Papineni K, Roukos S, Ward T, et al. Bleu: a method for automatic evaluation of machine tran slation[C]//Proceedings of the 40th annual meeting of the Association for Computational Li nguistics. 2002: 311-318.

Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30 (NIPS 2017), pp. 5998–6008 (2017)