Multiloop Incremental Bootstrapping for Low-Resource Machine Translation

Wuying Liu [1,2] wyliu@ldu.edu.cn
[1] Shandong Key Laboratory of Language Resources Development and Application, Ludong University, Yantai, 264025, China
[2] Center for Linguistics and Applied Linguistics, Guangdong University of Foreign Studies, Guangzhou, 510420, China
Wei Li 676471162@qq.com
School of Computer and Big Data Science, Jiujiang University, Jiujiang, 332005, China
Lin Wang * lwang@xdsisu.edu.cn
Xianda College of Economics and Humanities, Shanghai International Studies University, Shanghai, 200083, China
* Corresponding Author

Abstract

Due to the scarcity of high-quality bilingual sentence pairs, some deep-learning-based machine translation algorithms cannot achieve better performance in low-resource machine translation. On this basis, we are committed to integrating the ideas of machine learning algorithm improvement and data augmentation, propose a novel multiloop incremental bootstrapping framework, and design the corresponding semi-supervised learning algorithm. This framework is a meta-frame independent of specific machine translation algorithms. This algorithm makes full use of bilingual seed data of appropriate scale and super-large-scale monolingual data to expand bilingual sentence pair data incrementally, and trains machine translation models step by step to improve the translation quality. The experimental results of neural machine translation on multiple language pairs prove that our proposed framework can make use of continuous monolingual data to raise itself. Its effectiveness is not only reflected in the easy implementation of state-of-the-art low-resource machine translation, but also in the practical option to quickly establish precise domain machine translation systems.

1. Introduction

Machine Translation (MT) is an algorithmic computing process that uses a target natural language form to paraphrase the semantics of a source natural language. After the Bronze Age marked by Rule-based MT (RBMT) and the Silver Age marked by Statistical MT (SMT), the Golden Age marked by deep-learning-based Neural MT (NMT) has begun. After more than 70 years of unremitting exploration around the three generations of MT, many excellent algorithms and practical products have been produced (Garg and Agarwal, 2018).

If the formal language theory and context-free grammar derived from the development of compilers have achieved MT based on transformation generation rules, then language data has become the backbone of MT in the post-rule era. The Bayes conditional probability formula explicitly quantifies the language model and translation model contained in large-scale language data, which makes the noise channel model to decrypt an encrypted message become a

Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track, pages 1–11
September 4–8, 2023, Macau SAR, China.

statistical MT paradigm. The deep neural network performs fine-grained characterization of super-large-scale language data, and uses many parameters to simulate the end-to-end NMT model that can generate fluent target language (Tan, Wang, Yang, Chen, Huang, Sun and Liu, 2020).

RBMT is time-consuming and labor-intensive, and it is not easy to guarantee the self-consistency among many rules, so it is difficult to popularize into practical applications. SMT often needs more than 5 million sentence pairs to train a good model, while NMT requires at least 20 million sentence pairs to train an excellent model. The effect of NMT rolling that of RBMT and SMT is the result of the interaction of computing power, algorithm and data. It is precisely because the vector computing component has greatly accelerated the parallel computing ability, which makes the early proposed artificial neural network algorithm can burst out amazing deep intelligence on the data of super-large-scale bilingual sentence pairs (Stahlberg, 2020).

Among the more than 7,000 existing languages in the world, the vast majority of less commonly taught languages, such as indigenous languages, endangered languages, and dialects that are not widely spoken, have difficulties in data scarcity of super-large-scale bilingual sentences to varying degrees. Therefore, there is still huge room for improvement in low-resource MT with limited training data (Ranathunga, Lee, Skenduli, Shekhar, Alam and Kaur, 2021). At present, low-resource MT has gradually evolved into two mainstream research ideas, data augmentation centric idea and machine learning algorithm improvement centric idea. There is an overlap between the two ideas since the latter one may also use various language data.

2. Related Works

Reviewing the research history of low-resource MT, the data augmentation centric idea mainly focuses on how to expand the training corpus. While the machine learning algorithm improvement centric idea often explores how to use transfer learning, unsupervised learning, adversarial learning, and so on to improve the effect of low-resource MT.

Typical data augmentations include: **(1)** By pairing monolingual training data with an automatic back-translation, the approach can treat it as additional parallel training data, and obtain substantial improvements on the low-resource MT task (Sennrich, Haddow and Birch, 2016). **(2)** The method starts with a small amount of parallel data and iteratively improves the model by training it on the current data and using it to generate translations for additional monolingual data. (Hoang, Koehn, Haffari and Cohn, 2018). **(3)** Some studies use a bilingual lexicon to build a phrase-table, combine it with a language model, and use the resulting MT system to generate a synthetic parallel corpus, which does not require any additional resource besides the monolingual corpus used to train the embeddings (Artetxe, Labaka and Agirre, 2019).

Classical machine learning algorithm improvements include: **(1) Transfer learning**. The earlier technique is transfer learning between vocabulary, grammar and cognate languages mainly based on the characteristics of the language itself. Some studies first train a high-resource language pair (the parent model), then transfer some of the learned parameters to the low-resource pair (the child model) to initialize and constrain training (Zoph, Yuret, May and Knight, 2016). Then there are studies that relieve the vocabulary mismatch by using cross-lingual word embedding, train a more language-agnostic encoder by injecting artificial noises, and generate synthetic data easily from the existing data, so as to implement transfer learning between languages with different vocabulary and grammar (Kim, Gao and Ney, 2019). Some studies prove that the cognate parallel corpus can improve the low-resource language NMT effectively, which mainly depends on the morphological similarity and semantic equivalence between the cognate languages (Liu, Xiao, Jiang and Wang, 2018). Recent technique tends to adopt pre-trained models in related languages to bootstrap the training of a low-resource MT model. According to the language affinity, the research also found that the use of multi-round fine-tuning of highly related multiple high-resource language pairs can further improve the effect of low-resource MT (Maimaiti, Liu, Luan and Sun, 2019). Some studies have systematically compared multistage fine-tuning, and relevant experiments have confirmed that multi-parallel corpora are extremely useful, and their multistage fine-tuning can give 3~9 BLEU score gains over a simple one-to-one model (Dabre, Fujita and Chu, 2019). A study has proposed a XLNet based pre-training method, that corrects the defects of the pre-training model, and enhance NMT model for context feature extraction. Experimental results on minority languages to Chinese tasks show that the generalization ability and BLEU scores of this method are improved, which fully verifies the effectiveness of the method (Wu, Hou, Guo and Zheng, 2021). There are also studies aimed at two related very low resource Sorbian languages. On the one hand, the authors pretrain the German-Upper-Sorbian model using masked sequence to sequence objective and then finetune using iterative back-translation. On the other hand, they use final German-Upper-Sorbian model as initialization of the German-Lower-Sorbian model, and then the same vocabulary in the two languages is used in the further training of iterative back-translation (Khatri, Murthy and Bhattacharyya, 2021). **(2) Unsupervised learning**. This technique involves training a MT model without using any labeled data. Different from the unsupervised method in the above data augmentation, some studies have proposed a novel method to train a NMT system in a completely unsupervised manner, relying on nothing but monolingual corpus, which completely removes the need of parallel data (Artetxe, Labaka, Agirre and Cho, 2018). Some studies propose two knowledge distillation methods and empirically introduce a simple method to translate between thirteen languages using a single encoder and a single decoder, making use of multilingual data to improve unsupervised neural MT for all language pairs (Sun, Wang, Chen, Utiyama, Sumita and Zhao, 2020). Some studies add an adapter layer with a denoising objective on top of pre-trained model, and implement multilingual unsupervised MT that only has monolingual data by using auxiliary parallel language pairs (Üstün, Berard, Besacier and Gallé, 2021). **(3) Adversarial learning**. This technique adopts an interesting idea of alternate promotion of both two sides of contradiction. Some studies have put forward a unique idea of training the NMT model to generate human-like translations directly by using the generative adversarial net (Yang, Chen, Wang and Xu, 2018). The method builds a conditional sequence generative adversarial net which comprises of two adversarial sub models, a generative model which translates the source sentence into the target sentence as the traditional NMT models do and a discriminative model which discriminates the machine-translated target sentence from the human-translated one. The two sub models play a minimax game and achieve a win-win situation when reaching a Nash Equilibrium.

Overall, low-resource MT algorithms are still an active area of research, and there are many promising techniques being developed to improve the quality of translations for low-resource languages. We propose a novel multiloop incremental bootstrapping (MIB) meta framework independent of specific MT algorithms, and hope to integrate the advantages of data augmentation and machine learning algorithm improvements from a higher level of abstraction to achieve concise and efficient industrial practical methods.

3. Multiloop Incremental Bootstrapping

The MIB we proposed is a semi-supervised incremental learning data augmentation idea that can promote the advantages of supervised learning and unsupervised learning. The idea adopts a rolling snowball strategy: Firstly, good bidirectional MT models are trained by using bilingual corpus of appropriate scale. Then, through fully tapping the potential of the Internet monolingual big data, the trained MT models translate monolingual sentences twice to incrementally construct a bilingual pseudo-corpus. Then, the bilingual pseudo-corpus is used to enhance the initial bilingual corpus. Finally, the above process is loop-repeated based on the enhanced bilingual corpus, until the trained MT model meets the optimal performance requirements.

3.1. Framework

According to the MIB idea, we give full play to the advantages of super-large-scale unlabeled corpora, and propose a MIB framework for low-resource MT as shown in Figure 1. The framework mainly includes a MT model trainer, two machine translators, several crawlers, a similarity calculator, and a corpus truncator.

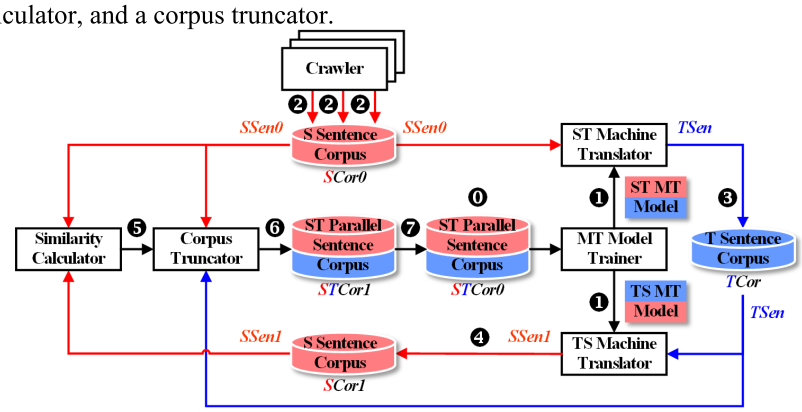

Figure 1. Multiloop incremental bootstrapping framework.

The MIB route is made up of multiple improvement loops. **Step⓿**: We need to prepare a ST (source language to target language) parallel sentence corpus named as *STCor0*. **Step❶**: The MT model trainer receives the *STCor0*, and trains out two MT models respectively from language S to language T and from language T to language S. **Step❷**: A group of parallel crawlers continuously crawl language S texts from the Internet, and build a super-large-scale language S sentence corpus (*SCor0*). **Step❸**: The ST machine translator translates each language S sentence (*SSen0*) in *SCor0* into the corresponding language T sentence (*TSen*) according to the ST MT model, and collects them to form a language T sentence corpus (*TCor*). **Step❹**: The TS machine translator translates the language T sentence (*TSen*) in *TCor* back into the language S sentence (*SSen1*) according to the TS MT model, and collects them to form a language S sentence corpus (*SCor1*). **Step❺**: The similarity calculator calculates the similarity between the source sentence *SSen0* and the result sentence *SSen1* flowing through the two machine translators. **Step❻**: The corpus truncator sorts the corresponding sentence pair <*SSen0*, *TSen*> according to the similarity between *SSen0* and *SSen1*, and truncate the TopN sentence pairs with the highest similarity to form a new ST parallel sentence corpus (*STCor1*). **Step❼**: The *STCor1* is merged into the *STCor0*. The first closed loop is completed from the Step⓿ to the Step❼, and then the second loop is started from the Step⓿ again, and so on. The above multiple loops are used together to implement the complete MIB framework.

Our MIB framework gives a novel idea of semi-supervised low-resource MT based on pseudo-corpus incremental learning. It has three significant features: **(1)** The framework is a very flexible meta-framework. On the one hand, it is independent of both specific MT model training algorithms and sentence similarity calculating algorithms. On the other hand, if a domain-independent universal parallel sentence corpus is used as the *STCor0*, and a directionally-crawled domain-dependent language S sentence corpus is used as the *SCR0*, it can quickly and conveniently implement precise MT systems adapting to various domains. **(2)** Although the working of the crawlers is a step contained in the loop, the preparation of the corresponding language S sentence corpus can also be separated out to establish an individual module. Because the scale of the language S sentence corpus affects the effectiveness of incremental learning, it is necessary to implement functions such as uninterrupted crawling, sentence segmentation, and sentence deduplication. We can use parallel computing technology to maximize the scale of the language S sentence corpus, use NLP technology to segment the language S sentences, and use information retrieval technology to delete the language S sentences contained in *STCor0*. **(3)** Two prior parameters need to be set. Where, the TopN parameter indicates the increment of sentence pairs in each loop, which determines the delta effect of each loop learning. The parameter of the total number of loops not only represents the MIB termination condition, but also determines the total learning time overhead. The two parameters together determine the depth of the whole learning.

3.2. Algorithm

According to the MIB idea, we design a multiloop incremental bootstrapping machine translation (MIBMT) algorithm as shown in the pseudo-code in Figure 2 to specifically implement the above MIB framework. The MIBMT algorithm mainly includes two main functions of MIB training (MTMODELS: ***train*** ()) and translating (STRING: ***translate*** ()), and a specific model training function (MTMODEL: ***modeltrain*** ()), a crawling function (SCOR: ***crawl*** ()), and so on.

```
// Multiloop Incremental Bootstrapping Machine Translation (MIBMT) Algorithm
// MIB Training
// n: total number of loops
// topn: increment of sentence pairs
// stcor0: parallel sentence corpus
Main Function MTMODELS: train(n, topn, stcor0)
MTMODELS mtmodels ← MTMODELS.new();
For 0 To n Do
    mtmodels.st ← modeltrain(stcor0, 's', 't');
    mtmodels.ts ← modeltrain(stcor0, 't', 's');
    SCOR scor0 ← crawl(stcor0.get('s'), 's');
    STCOR stcor1 ← STCOR.new();
    For STRING ssen0 : scor0 Do
        STRING tsen ← translate(mtmodels.st, ssen0, 's');
        STRING ssen1 ← translate(mtmodels.ts, tsen, 't');
        FLOAT sim ← similaritycalculate(ssen0, ssen1);
        stcor1← corpustruncate(stcor1, ssen0, tsen, sim, topn);
    End For
    stcor0.merge(stcor1);
    mtmodels ← MTMODELS.new();
End For
Return mtmodels.
// Specific Model Training
// stcor0: parallel sentence corpus
// ls: source language id
// lt: target language id
Function MTMODEL: modeltrain(stcor0, ls, lt)
SCOR newscor ← mpt.tokenize(stcor0.get(ls), ls);
TCOR newtcor ← mpt.tokenize(stcor0.get(lt), lt);
MTMODEL mtmodel ← specificmodeltrain(newscor, newtcor);
Return mtmodel.
// Crawling
// scor: sentence corpus
// l: language id
Function SCOR: crawl(scor, l)
SCOR scor0 ← SCOR.new();
SCOR crawledscor ← mpt.sensplit(crawledtxt, l);
For STRING sen : crawledscor Do
    If (!scor.contain(sen)) Then scor0.add(sen);
End For
Return scor0.
// MIB Translating
// mtmodel: machine translation model
// inputtxt: input text
// ls: source language id
Main Function STRING: translate(mtmodel, inputtxt, ls)
STRING outputtxt ← STRING.new();
SCOR inputscor ← mpt.sensplit(inputtxt, ls);
For STRING sen : inputscor Do
    outputtxt ← outputtxt + mtmodel.specifictranslate(sen) + separator;
End For
Return outputtxt.
```

Figure 2. Multiloop incremental bootstrapping machine translation algorithm.

In the main function of MIB training (Function MTMODELS: *train* ()), the inputs are the preset total number of loops (*n*), increment of sentence pairs (*topn*), and initial parallel sentence corpus (*stcor0*), while the output is a pair of MT models (*mtmodels*). The outermost loop is run n+1 times based on the preset total number of loops n (lines 8 to 21 in Figure 2). In each loop, firstly, perform bidirectional translation model training (lines 9 and 10 in Figure 2), then crawl the monolingual sentence corpus (line 11 in Figure 2), then perform bidirectional translation on the sentences in the monolingual sentence corpus one by one, and obtain the pseudo bilingual sentence corpus according to the similarity (lines 13 to 18 in Figure 2). Finally, merge the pseudo corpus into the initial parallel sentence corpus.

In the main function of MIB translating (**Function** STRING: ***translate*** ()), the inputs are MT model (*mtmodel*), source language text (*inputtxt*), and source language identifier (*ls*), while the output is target language text (*outputtxt*). Firstly, the input source language text is segmented into sentences (line 48 in Figure 2), then translated one by one according to the translation model (line 50 in Figure 2), and the translated sentences are connected and assembled by the target sentence separator (loop between lines 49 and 51 in Figure 2), finally the target language text is output.

The inputs of the specific model training function (**Function** MTMODEL: ***modeltrain*** ()) are a parallel sentence sorpus (*stcor0*), a source language identifier (*ls*), and a target language identifier (*lt*). The output is a translation model (*mtmodel*). In addition to the need for token feature representations based on language (lines 28 and 29 in Figure 2), the most important step is the specific training step for MT models (line 30 in Figure 2), which is an end-to-end training process for NMT models.

The inputs of the crawling function (**Function** SCOR: ***crawl*** ()) is the existing monolingual sentence corpus (*scor*) and language identifier (*l*), while the output is the newly added monolingual sentence corpus (*scor0*). In addition to sentence segmentation for the crawled text (line 37 in Figure 2), it is necessary to perform a repeat judgment operation (line 39 in Figure 2) to ensure that the new sentence is not in the existing monolingual sentence corpus.

3.3. Algorithm Analysis

Inheriting the meta-framework characteristic of the MIB framework, the MIBMT algorithm is also a general meta-algorithm. Any high-performance MT algorithm can be embedded in the meta-algorithm to implement specific functions of model training and translating. This meta-algorithm uses repetitive hardware multi-process and multi-threading to implement the efficient crawling (***crawl***), and uses sentence fingerprint indexing and retrieval to implement the Boolean judgment (***contain***). There are three characteristics that need special attention in practical use: (i) The scale and quality of the initial parallel sentence corpus *stcor0* must meet the requirements of specific model training to ensure that the MT model trained in the first loop has high translation precision. Just as “no powerful First Impulse, no more and more precise celestial orbits”. (ii) The MIBMT bias is controlled by the crawled super-large-scale sentence corpus *scor0*. If *scor0* comes from open domain contents, a universal MT model is eventually produced, while from narrow domain contents, a domain MT model is produced. “What foods he feeds, what eggs hen will lay.” (iii) For each source language or target language, a dedicated morphological processing tool (***mpt***) is required. For instance, during specific model training, the tokenize tool (***tokenize***) represents each single Chinese character as a token, while it represents the lowercase form of each English word separated by spaces as a token. For another instance, both the crawled text (*crawledtxt*) and input text (*inputtxt*) need to execute a sentence splitting tool (***sensplit***) according to the corresponding language to obtain a sentence sequence. We have to customize the morphological processing tool for each language because different languages have different morphological representation systems. That is “different shoes for different feet”.

The time overhead of the MIBMT algorithm is mainly used for the learning process of the training function, which is directly proportional to the total number of loops and the training time of the specific MT model. For instance, the total number of loops is n, the time cost of training a specific model using a NMT algorithm is m, and the bidirectional models are trained in parallel (line 9, 10 of Figure 2), then the main time complexity will be $\boldsymbol{O}(nm)$. The space overhead of the MIBMT algorithm is not only related to the increment of sentence pairs (TopN) and the size of the initial parallel sentence corpus, but also to the space cost of the specific MT model. Since the training corpus is processed in batches during model training, this relationship is only a positive correlation and not a simple direct proportional relationship. If a NMT model is specifically used, and the source language vocabulary size is S and the target language vocabulary size is T, then the main spatial complexity is $\boldsymbol{O}(ST)$. Of course, the above-mentioned space-time complexity is still very huge. Fortunately, the learning process of the training function is an offline processing, and it is learned once and used multiple times. While the online processing of the translating function is efficient in time and space. In order to achieve excellent MT results in practical applications, longer learning time and larger storage space are worthwhile and acceptable. We can also increase the GPU and memory to reduce actual space-time overhead.

4. Experiment

In order to verify that the MIB can be effectively used for low-resource MT, we first implement a MIBMT meta-algorithm by embedding an open source sequence-to-sequence NMT model[1]. The hparams of the NMT model mainly include the number of neurons (*num_units* = 512), the number of encoding and decoding layers (*num_encoder_layers* = *num_decoder_layers* = 4), the batch size (*batch_size* = 512), and the beam search width (*beam_width* = 10), while others remain the default values. Next, the 15 languages of Indonesian (ind), Malay (msa), Vietnamese (vie), Thai (tha), Khmer (khm), Lao (lao), Filipino (fil), Myanmar (mya), Italian (ita), Kazakh (kaz), Kyrgyz (kir), Ukrainian (ukr), Polish (pol), Czech (ces), and Slovak (slk), which are relatively scarce in parallel sentence pair resources to Chinese (zho), are selected and their morphological processing tools are implemented respectively. Finally, a prototype system for MT experiments from these 15 languages to Chinese was built.

A total of 15 NMT models need to be trained to support MT from the 15 low-resource languages to Chinese in the experimental prototype system, which have been successfully deployed as web application systems at present[2]. During the training of these models, we fixed the total number of loops and the increment of sentence pairs (TopN) to 11 and 1,000,000 respectively. The parallel sentence corpus (*STCor0*) for each language and Chinese, that is, the initial training set, contains 5,000,000 sentence pairs, while the final training set will contain 15,000,000 sentence pairs after the 11 loop executions. At the same time, in order to train specific sequence-to-sequence NMT models, we also equip an additional 100,000 sentence-pair development set and 100,000 sentence-pair test set for each language. For each language, the initial training set is exactly the same distribution as the development set and the test set, which are divided from the same corpus by simple random sampling. While the crawler captures from open domain to form the monolingual sentence corpus (*SCor0*), which is independent of the initial training set. In order to ensure the high availability of the Top1,000,000 pseudo-corpus, monolingual sentences at least 10 times TopN is captured in each loop, and then the Top1,000,000 sentence pairs are truncated based on the Levenshtein similarity.

---

1 https://github.com/tensorflow/nmt

2 http://nmt.cpolar.cn

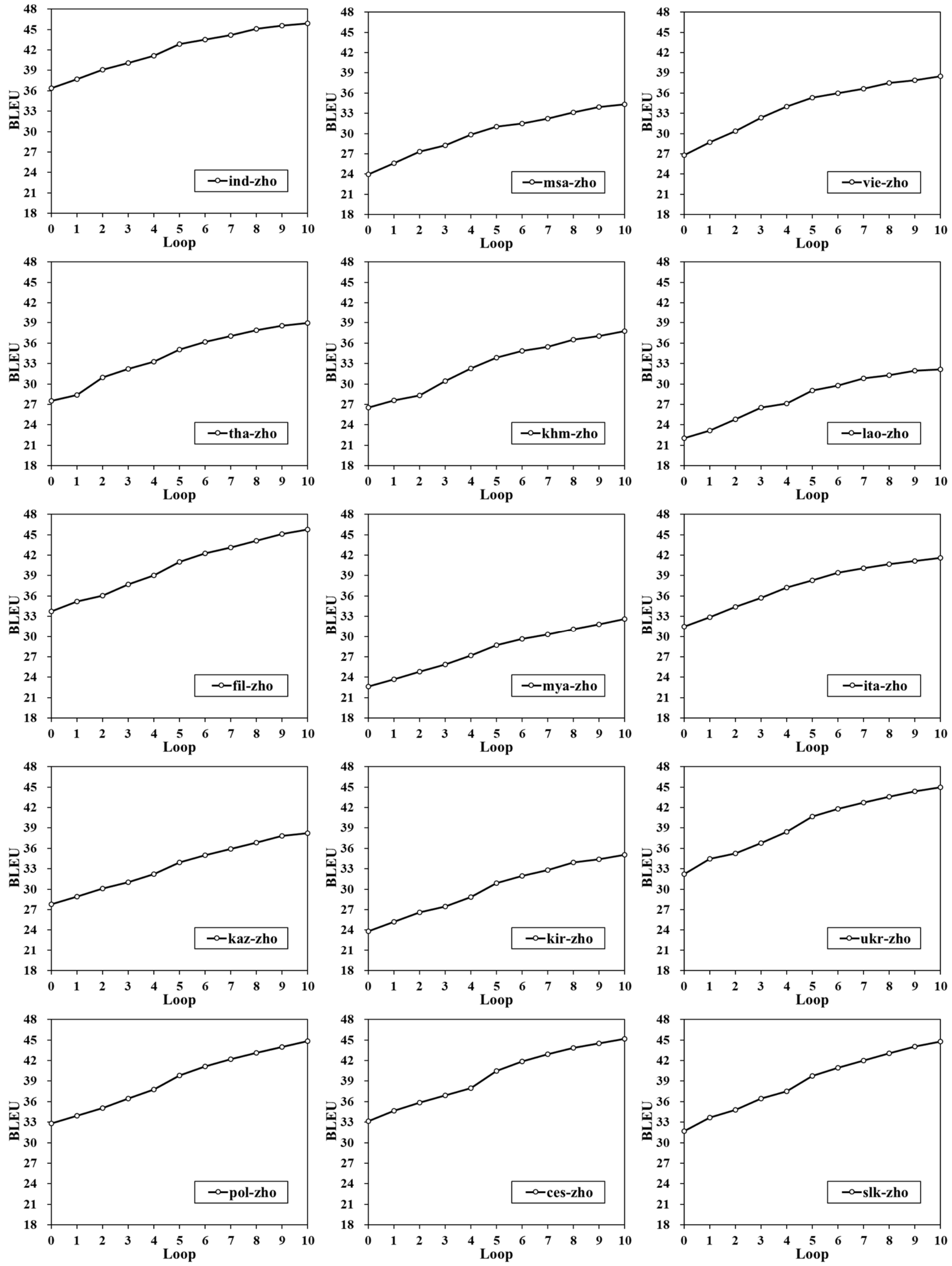

Figure 3. BLEU trend curves.

After 15 months of training, the BLEU trend curves of the above 15 MT models are shown in Figure 3. Where, the abscissa axis represents the loop ordinal, and the ordinate axis represents the BLEU value. Among them, Loop=0 represents the sequence to sequence NMT model obtained from the initial training set of 5,000,000 sentence pairs without pseudo corpus, which is used as the benchmark model for the following effect comparison. We find from the curves in Figure 3: (i) The BLEU value of each curve increases approximately linearly with the number of loops (increment of 1,000,000 sentence pairs per loop). This shows that the MIB has a general promotion effect on low-resource MT lack of bilingual sentence pair resources. The reason is that with the extension of the corpus, the vocabulary space is more complete and the model is more generalized. (ii) Almost the linear growth rate of each curve around the Loop=5 point will change slightly, and the linear growth rate in the first half is greater than that in the second half. Among them, the vie-zho curve is the most obvious. This shows that when the scale of real corpus is larger than that of pseudo-corpus, the enhancement effect of pseudo-corpus is more significant. Because the fixed test set often has the best fit model, when the proportion of the pseudo-corpus is too large, it may cause overfitting. (iii) There is a significant difference among the BLEU values of the initial model Loop=0 in different languages, with a maximum difference of over 10, while the BLEU increment (ΔBLEU) between the final model and the initial model is basically maintained at around 10. This is because different languages have different entropy, so the information contained in the same scale corpus is not equal, resulting in uneven performance of the initial model trained by sentence pairs of the same scale.

Source-Target Language	Vocabulary Size of Source Language of Loop 10	Vocabulary Size of Target Language of Loop 10	BLEU of Loop 10	ΔBLEU between Loop 10 and Loop 0
ind-zho	604,869	8,960	**45.90**	9.55
msa-zho	357,264	8,168	34.32	10.38
vie-zho	66,242	8,293	38.51	11.71
tha-zho	8,110	6,980	38.95	11.45
khm-zho	128,930	6,995	37.77	11.22
lao-zho	149,478	6,913	32.12	10.07
fil-zho	201,835	6,393	**45.74**	12.01
mya-zho	24,384	6,907	32.60	9.93
ita-zho	884,503	9,759	41.57	10.11
kaz-zho	699,425	7,017	38.26	10.44
kir-zho	740,651	7,007	35.03	11.18
ukr-zho	627,365	7,023	44.94	12.69
pol-zho	541,620	6,929	44.85	12.04
ces-zho	550,807	6,931	**45.14**	12.02
slk-zho	576,679	6,930	44.79	**13.11**

Table 1. Final vocabulary size and BLEU values.

The final vocabulary size and corresponding BLEU values are shown in Table 1. Where, the Chinese vocabulary size is relatively fixed, with value ranging from 6,000 to 10,000. Each “word” in the Chinese vocabulary is a single Chinese character or other token. But there are two forms of uppercase and lowercase in Latin, Cyrillic or other alphabet languages, which use a lowercase vocabulary for MT to Chinese. Observing the final BLEU values, we found that the BLEU value of the NMT model from Indonesian, Filipino and Czech to Chinese exceeded 45. Among them, the BLEU value of Indonesian-Chinese NMT model reaches the highest of 45.90. Observing the ΔBLEU values between Loop=10 model and Loop=0 model, it is found that the BLEU values of 15 low-resource languages to Chinese NMT models can be improved between 9.55 and 13.11 by using the proposed method. The BLEU value of the Slovak-Chinese NMT model increased the most, while that of the Indonesian-Chinese NMT model increased the least. It can be seen that the higher the performance of Loop=0 model, the higher the final performance can be obtained by adopting the MIB method. In summary, the experimental results prove that our proposed MIB is effective for low-resource MT.

5. Conclusion

The incremental pseudo-corpus in the MIB of this paper is derived from the newly trained MT models, while the MT models are trained from the training set enhanced by newly produced pseudo-corpus, which is a closed-loop self-lifting idea based on the homogeneous MT models. The experimental results on multiple languages prove that the language resources expanded by this idea can effectively improve the performance of low-resource MT.

The next research will concern an open-loop mutual-lifting idea based on heterogeneous MT models. That is, the source and output MT models of incremental pseudo-corpus are two different excellent MT models. It is hoped that an excellent MT model will enhance another one through the produced corpus transmission. In addition, we also hope to transfer the general MIB framework of this paper to low-resource MT in other languages and precise domain MT.

Acknowledgments

The research is supported by the New Liberal Arts Research and Reform Practice Project of Ministry of Education of China (No. 2021060049), the Postgraduate Education and Teaching Reform Research Project of Shandong (No. SDYJG21185), the Key Project of Undergraduate Teaching Reform Research of Shandong (No. Z2021323), the Science and Technology Program of Guangzhou (No. 202201010061), the Humanity and Social Science Research Project of Ministry of Education of China (No. 20YJAZH069, No. 20YJC740062), and the Social Science Foundation of Shanghai (No. 2019BYY028).

References

Garg, A. and Agarwal, M. (2018). Machine Translation: A Literature Review. arXiv:1901.01122v1.

Tan, Z., Wang, S., Yang, Z., Chen, G., Huang, X., Sun, M., and Liu, Y. (2020). Neural Machine Translation: A Review of Methods, Resources, and Tools. *AI Open*, 1:5–21.

Stahlberg, F. (2020). Neural Machine Translation: A Review. *Journal of Artificial Intelligence Research*, 69:343–418.

Ranathunga, S., Lee, E. A., Skenduli, M. P., Shekhar, R., Alam, M., and Kaur, R. (2021). Neural Machine Translation for Low-Resource Languages: A Survey. arXiv:2106.15115v1.

Sennrich, R., Haddow, B., and Birch, A. (2016). Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96, Berlin, Germany.

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

Artetxe, M., Labaka, G., and Agirre, E. (2019). Bilingual Lexicon Induction through Unsupervised Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5002–5007, Florence, Italy.

Zoph, B., Yuret, D., May, J., and Knight, K. (2016). Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, USA.

Kim, Y., Gao, Y., and Ney, H. (2019). Effective Cross-lingual Transfer of Neural Machine Translation Models without Shared Vocabularies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1246–1257, Florence, Italy.

Liu, W., Xiao, L., Jiang, S., and Wang, L. (2018). Language Resource Extension for Indonesian-Chinese Machine Translation. In *Proceedings of the 22nd International Conference on Asian Language Processing*, pages 221–225, Bandung, Indonesia.

Maimaiti, M., Liu, Y., Luan, H., and Sun, M. (2019). Multi-Round Transfer Learning for Low-Resource NMT Using Multiple High-Resource Languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 18(4):1–26.

Dabre, R., Fujita, A., and Chu, C. (2019). Exploiting Multilingualism through Multistage Fine-Tuning for Low-Resource Neural Machine Translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1410–1416, Hong Kong, China.

Wu, N., Hou, H., Guo, Z., and Zheng, W. (2021). Low-Resource Neural Machine Translation Using XLNet Pre-training Model. In *Proceedings of the 30th International Conference on Artificial Neural Networks and Machine Learning*, pages 503–514, Bratislava, Slovakia.

Khatri, J., Murthy, R., and Bhattacharyya, P. (2021). Language Model Pretraining and Transfer Learning for Very Low Resource Languages. In *Proceedings of the 6th Conference on Machine Translation*, pages 995–998, Online.

Artetxe, M., Labaka, G., Agirre, E., and Cho, K. (2018). Unsupervised Neural Machine Translation. In *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada.

Sun, H., Wang, R., Chen, K., Utiyama, M., Sumita, E., and Zhao, T. (2020). Knowledge Distillation for Multilingual Unsupervised Neural Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3525–3535, Online.

Üstün, A., Berard, A., Besacier, L., and Gallé, M. (2021). Multilingual Unsupervised Neural Machine Translation with Denoising Adapters. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6650–6662, Online and Punta Cana, Dominican Republic.

Yang, Z., Chen, W., Wang, F., and Xu, B. (2018). Generative Adversarial Training for Neural Machine Translation. *Neurocomputing*, 321:146–155.