Neural Machine Translation in Translation and Program Repair

Tingsong Huang[1, *], Yifei Jia[2] , Haohua Pang[3] and Zhe Sun[4]

[1] College of Computer and Data Science, Fuzhou University, Fuzhou, Fujian, 350108, China
[2] Jinan-Birmingham Joint Institute, Jinan University, Guangzhou, Guangdong, 511436, China
[3] Division of Science and Technology, Beijing Normal University - Hong Kong Baptist University United International College, Zhuhai, Guangdong, 519087, China
[4] School of Business and Management, Jilin University, Changchun, Jilin, 130015, China

041901413@fzu.edu.cn

Abstract. Translation is a challenge for humans since it needs a good command of two or more languages. When it comes to computer programs, it is even more complex as it is difficult for computers to imitate human translators. With the emergence of deep learning algorithms, especially neural network architectures, neural machine translation (NMT) models gradually outperformed previous machine translation models and became the new mainstream in practical machine translation (MT) systems. Nowadays, NMT has been developing for several years and has been applied in many fields. This paper is focused on studies on four different application categories of NMT models: 1) Text NMT; 2) Automatic program repair (based on NMT); 3) Simultaneous translation. Our work provides a summary of the latest research on different applications of NMT and makes comments on their development in the future. This paper also mentioned the shortcomings of existing studies in this essay and pointed out some possible research directions.

Keywords: neural machine translation, text NMT, automatic program repair.

1. Introduction

MT is a translation method that uses machine learning techniques to translate written or spoken content from one language to others. However, because of the complexity and flexibility of natural language, MT is a challenging task. Statistical machine translation (SMT), as one of the solutions to this task, was firstly proposed in 1949 and re-introduced in the late 1980s and early 1990s. SMT generates translation by learning latent structure directly from bilingual text corpora. Due to the lack of long-term dependency between words, the translation output by SMT is not accurate enough.

Up until a few years ago, with the current breakthrough of deep learning, a deep neural network model-based approach to MT, which is aptly named neural machine translation (NMT), got the highest development, and rapidly became the new mainstream in practical MT systems. NMT adapts artificial neural network architectures to learn a statistical model for MT. The dissimilarity between SMT and

NMT is that NMT attempts to train a single system that can learn the oral or written language and output a correct translation instead of requiring sub-components that are tuned separately [1]. Therefore, NMT is said to be an end-to-end system.

Recently, many studies are focused on the different applications of NMT. This paper presents a survey of the results in these applications of NMT. This review describes the commonly used modeling techniques, the different categories of applications, and relevant works on them. The first part of the review introduces text NMT by analyzing its sub-fields: word/short sentence-level NMT, character-level NMT, document-level NMT, and low-resource NMT. This part highlights some relative works on these tasks and makes comments on them. The second section is mainly about automatic program repair (based on NMT), which is a newly emerged application of NMT. Then, the article summarizes the development of simultaneous translation, which is about the attempts to interpret oral contents. Finally, this paper discusses the challenges, future work recommendations, and conclusions of these applications of NMT models.

2. Overview

Starting from 2014, the models of NMT have emerged one after another, and the highest state-of-art has been constantly updated. However, as of 2022, most of the new models are based on original basic NMT models, such as encoder-decoder approaches and attentional mechanisms.

2.1. Encoder-decoder approaches

Encoder-decoder is a very common kind of frameworks in deep learning. The structure that NMT with Encoder-decoder approaches depends on is sequence to sequence, which is composed of two RNN. One RNN is used as an encoder, which makes the input source language transformed into a vector in a representation space, and the other RNN acts as a decoder to convert it into sentences in the target language. In this architecture, Decoder can be regarded as a language model to predict the next word of the target language, and its probability depends on the encoder of the source language. In practical application, encoder and decoder can be not only simple RNN, but also double-layer RNN or LSTM.

Generally, compared with SMT that requires Bayesian theorem, NMT can directly calculates.

$$p(y|x) = p(y_1|x)p(y_2|y_1, x)p(y_3|y_1, y_2, x)\ldots p(y_T|y_1, y_2, \ldots, y_{T-1}, x) \quad (1)$$

And as shown in Figure 1, the loss function, which is the cross entropy of each decoder hidden layers, is

$$J(\theta) = \frac{1}{T}\Sigma_{t=1}^{T}J^{(t)}(\theta) = \frac{1}{T}\Sigma_{t=1}^{T} - log\hat{y}_{x_{t+1}}^{(t)} \quad (2)$$

The gradient of the loss function can be propagated back to the encoder, and the model can be optimized. So seq2seq is also regarded as the end2end model, which means this model only cares about the input and output and ignores all the intermediate steps. And it emphasizes the global optimization rather than the local optimal solution of the greedy algorithm.

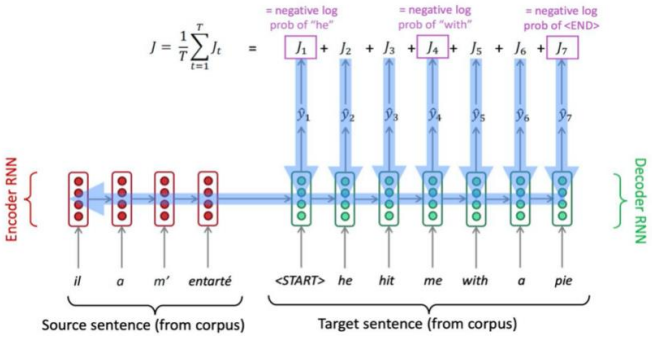

Figure 1. Encoder-Decoder architecture and corresponding loss function.

This work used the encoder RNN to summarize the input statement information into the last hidden vector. And then it can continue to let the last encoder hidden vector as the first decoder hidden vector to decode the text in the corresponding other languages. However, after the input long sequence is transformed into a fixed length vector, all the effective information cannot be saved, so the effect of this structure will decrease significantly with the increase of the length of the translated sentence.

2.2. Attentional mechanism

In the above, it can see that the seq2seq model has two bottlenecks: (1) The first one is the capacity bottleneck of the coding vector, that is, all the information of the source language needs to be stored in the coding vector to be decoded effectively. (2) The second one is the problem of long-distance dependence, that is, in long-term transmission the information will lose in the encoding and decoding networks.

By introducing the attentional mechanism, it can solve the problems by saving the information about every location in the source language. When generating each word of the target language in the decoding process, it directly selects the relevant information from the information of the source language through the attentional mechanism.

3. Overview

3.1. Text NMT

As the name suggests, the basic use of NMT is to translate various types of text, ranging from long monolingual essay to short advanced mathematical equations with professional symbols. And for various types of text with different length, the methods that should be used are quite different in term of model improvements from original NMT methods according to distinct assumptions and text backgrounds.

For example, the simplification of advanced mathematical equations with symbolic inference can be solved by a NMT model with an added special mechanism which will never be used in neural machine-translation for colloquialisms and idioms.

3.1.1. Word/Short sentence-level NMT. Word-level and short sentence-level NMT is the first and earliest NMT. Originally, word-level NMT usually adopts encoder decoder architecture for modeling, which is composed of embedding layer, classifier layer, encoder network and decoder network. In 2014, Kyunghyun Cho and his teammates analyzed an RNN Encoder-Decoder and a gated recursive convolutional neural network, and they showed that the performance of NMT with RNN Encoder-Decoder was good on not long sentences with a complete dictionary [2]. However, its performance decreased significantly if the sentences' length and the number of unknown words rose. After Cho's works, the author of presented an advanced encoder-decoder architecture for the NMT model. They found that the word-vector with fixed length was a check point in enhancing the work of the original encoder-decoder architecture. And they extended this architecture by proposing a new model which can search for parts of a source sentence automatically without forming these parts as a fixed partition. In another study, the authors developed an attentional mechanism which can promote the NMT performance by selectively picking some useful segments of the source sentence during the process [3]. In this study, they also used some datasets to test two classical attentional mechanism. One was a global approach which will pay attention to all source vocabulary, and the other was a local approach which will just search a subset of source vocabulary once.

Especially, mathematical expression, the most important and common thing in science articles, can be also considered as a type of "text". In the work of, the authors transform mathematical expressions into tree structures, and then train the network to find clues about solutions among symbols. Finally, the simplified expressions of ordinary equation even with integral are deduced by a seq2seq transformer model.

Since word-level NMT was proposed in 2014, researchers have continuously improved the seq2seq model, including the introduction of attentional model, the use of external memory mechanism, the use of semi-supervised learning and the modification of training criteria. In just two years, the performance of word-level NMT has exceeded the traditional phrase-based machine translation methods.

Although word-level NMT is so powerful, there are still some problems which include. First, there is not a flawless word segmentation algorithm for a language, which can divide any sentence into sequences composed of lexemes and morphemes. Then, there are rare words for most dataset. The problem with rare words is that the words in some dictionaries appear too few times in the training set, which makes it impossible to train a good word vector. And there are unknown words for most dictionary. Unknown word problem means that words that are not in the dictionary which are marked as UNK (unknown) or OOV (out-of-vocabulary).

3.1.2. Character-level NMT. For those above problems in word-level NMT, especially for the OOV problem, some scientists developed character-level NMT to overcome them with some innovative ideas and improved methods. In another work, the authors showed a seq2seq model whose innovation is mainly a new network structure, bi-scale RNN, on the decoder side [4]. This new bi-scale RNN is introduced to capture the information on the character and word timescale, to translate character sequences without word segmentation. Aiming to solve the OOV problem not only of source language but also of the target language, the author proposed a mixed word-character NMT model whose objective function of model training is cross-entry loss, considering the loss of word level and character level [5]. This new NMT model can perform well without too high training difficulty and model complexity. In addition, because of the addition of character-level processing, it is especially suitable for languages with words having rich variations. The author applied the method of character-based word embedding to NMT, which was able to overcome the problem of OOV in some cases [6]. At the same time, due to the use of the internal information of words, the method can also produce better outputs for the translation of languages with rich morphological changes. However, the author only used the above method in the source part. For the target part, they still faced the limitation of dictionary size. The authors also added character to word (c2w) and word vector to character (V2C) modules to the front and back of the attention based NMT model. As a character set model, it can naturally learn the relationship between prefixes and prefixes in some languages in translation. In addition, the model based on character level has flexibility in translating unknown words. The innovation of the next paper is to propose a text representation unit between characters and words and use The PE compression algorithm for reference to achieve a more balanced state in terms of vocabulary size and text length [7]. However, whether it can have similar effects on non-homologous / near source language pairs (such as French and Chinese) remains to be studied. And there is still room for discussion on the optimization of this NMT model.

To sum up, according to these advanced character-level NMT models, the main advantages of the character-level NMT include not being affected by the morphological changes of the language, being able to predict unknown words which are not in the dictionary, and downsizing vocabulary.

3.1.3. Document-level NMT. Wang and his teammates proposed a document-level RNN-based NMT model, which is significantly improved compared to the document independent sentence based NMT model [8]. After that, Bawden et al. proposed the NMT model with multiple encoders which can make use of the context of the previous source statement, to use the connection, gating, or hierarchical attention mechanism to get more accurate information by combining the earlier text and the present sentence [9]. In the work of extending the translation unit through context, Rios focused on word sense disambiguation (WSD) in NMT [10]. The way to solve this problem was to take the lexical chain of semantically similar words as the feature input of the network. Although this method did not produce substantial improvement compared with the baseline of the general test set, it had some improvement for the target test sets introduced in the same work by using previous models. It is worth mentioning that they also found evidence that without document-level context, even humans cannot eliminate some ambiguities in the target test set they use.

Next, in another study, scientists found a context-aware decoder that can be used in a basic transformer model. They allowed the multi-head attention sublayer to not only use the present sentence, but also use the earlier context. Their model performed well in the targeted test set introduced, but as far as BlEU is concerned, its effect was equivalent to that of the previous sentence-level translation model.

So far, all the mentioned works had used neural architecture, which combined contextual information by modifying the structure in the NMT model at the basic sentence level. However, some context information was useless, and it should be removed to promote the model's performance. Jean and CHO studied this problem from the perspective of learning and designed a regularization relationship to make the improved NMT model use context information in an effective way [11]. This method was suitable for token, sentence, and corpus levels, and it could make the transformer model more sensitive to other contexts. And In Bleu score, the promoted model with this new method was better than the original context independent transformer model.

3.1.4. Low-resource NMT. NMT techniques require sizable parallel corpora to obtain an accurate translation. However, there are more than 7000 languages all over the world but only a few of them have abundant parallel corpora. In this case, many methods based on NMT are proposed to make an advanced improvement on low-resource translation. Techniques could be divided into two classes: (i)Data augmentation techniques, and (ii) NMT improvement.

Data augmentation is a kind of strategy used to make more data from existing data or additional data to train NMT models. Parallel data extraction techniques are quite effective to achieve the goal. In these techniques, firstly a multilingual embedding representation is learned. Most of the early research in this field adapted supervised encoder-decoder architecture to generate multilingual embedding representation, which has the drawback that they require sizable parallel corpora for training [12]. Therefore, recent works proposed unsupervised NMT architectures instead [13]. After generating multilingual embedding, when given a sentence in one language, parallel sentences from other languages could be found.

Another important data augmentation technique is back translation. In back translation, sentences on the target side are translated reversely into the source language [14]. Then the generated synthetic parallel corpora as well as the original ones are used to train an NMT model. However, the data obtained by these techniques are usually noisy and inaccurate. As a result, iterative back-translation is proposed, where the target-side sentences are translated to the source side and then forward-translated to the target side repeatedly until there is no improvement on both sides [15]. Word or phrase replacement techniques are also used. In these techniques, new synthetic sentences are obtained by replacing words from a subset of sentences from an existing bilingual or monolingual corpus [16].

Another idea to obtain more accurate translation performance in the low-resource case is to improve existing NMT architectures. Unsupervised NMT assumes that there is no parallel data and thus only uses monolingual data. On the contrary, semi-supervised NMT combines monolingual corpora and small parallel corpora. Semi-supervised NMT can be classified into different categories according to where the monolingual data is used. Many recent works concentrate on dual learning, where the monolingual data is used as a reward signal [17]. Multi-NMT is proposed to solve the translation between multiple language pairs. Most of the multi-NMT are based on supervised NMT, while some are based on unsupervised or semi-supervised NMT. The transfer learning technique for low-resource NMT is to use the parent model, which is an NMT model learning a large parallel corpus extracted from abundant-resource language, and then fine-tune the model parameters on the target low-resource language to get the child model [18]. Warm-start transfer learning techniques, where the parent model also learns child parallel data, outperform the cold-start systems which means only high-resource parallel data is used when training the parent model. Fine-tuning methods with low-resource language data and one similar rich-resource language data are better than the ones with low-resource language data only [19].

Pivot translation (also known as pivoting) is a promising solution to zero-shot translation, which is a problem in that the model is trained under the circumstance where the model is trained without parallel data. In pivot translation, an intermediate rich-resource language acts as a 'bridge', which is called the 'pivot language'. Then, the source-target translation model is constructed based on source-pivot and pivot-target corpora.

3.2. Automatic program repair

Automatic program repair (APR) is vital in enhancing software reliability. The high performance of NMT on APR tasks is quite impressive. Automated generate-and-validate (G&V) program repair method is the major approach used for APR. The G&V method collects candidate patches generated from transformation or mutation and then ranks the patches via compiling and running a given test suite. Next top-ranked patches are returned. Based on the G&V method, NMT models feature strong learning capability for complicated relations between input and output sequences. Meanwhile, NMT models require less manual effort for different programming languages than vanilla G&V methods. The main reason is that the retraining process is automated and avoids implementing strictly fixed patterns. The advanced APR approaches based on NMT techniques can be introduced in two types. The APR related paper were organized into Table 1.

3.2.1. Context-aware Strategy. The two key restrictions about APR problems are what the NMT techniques need to overcome. The correct fix may not be included in the search space is a major challenge, and the other limitation is that fixing bugs need to follow strict code syntax. The advent of context-aware architecture aimed to resolve the limitations. As mentioned above, correct fixes may not be contained in the search space leading to the enlargement of search space which causes a decrease in repair effectiveness. Context-aware architecture works to seize information about the repair operations, correct fix components, and the hidden relations between them in terms of context. In the first study, a Context-Aware Patch Generation (CAPGEN) approach based on the Abstract Syntax Tree (AST) was presented [20]. The method, which aims to approach the search space via fix space, extracts fixing ingredients from context information and then integrates the factors to generate the prioritized candidate patches via the AST structure. Though the number of generated patches is not numerous, the method's performance on real bugs achieved high precision among the plausible patches. In another article, an architecture named SEQUENCER is presented [21]. The architecture generates one-line fixes to fix bugs. The fundamental structure of the NMT models used in the structure is the typical encoder-decoder system. To generate fixes via the buggy context transition, the sequence-to-sequence network based on RNNs with long short-term memory (LSTM) gates is used in the architecture. The precision is considerable though the generated number of the patches is not many either. The next study presented a new G&V technique called CoCoNuT [22]. Using the encoder-decoder mechanism, the new NMT architecture in CoCoNuT has two separate encoders to process the buggy information and context information so that the strong and valuable relations between tokens in the buggy contexts and the correct fixes can be extracted. Then the CNN (convolutional neural network)-based ensemble learning which associates models with diverse levels of complexity, enables the model to learn multiple repair strategies. CoCoNuT has shown impressive performance for APR for both the fixing precision and its capability of fixing bugs in a wide variety of programming languages. The number of generated patches is appreciable compared to the previous studies.

3.2.2. Code-aware Strategy. Context-aware strategy based on NMT can mine the relation between the context and buggy lines, but the restricted vocabulary and search spaces still cause limitations. All the correct fixes cannot be included in the search space with finite vocabulary size. CoCoNuT, which was mentioned above, uses over 130,000 tokens. However, there is still the out-of-vocabulary (OOV) problem, which also leads to the missing problem of search spaces. A large amount of generated uncompilable patches is another challenge for the context-aware models. Meanwhile, previous architectures failed to learn the patterns that developers write code. On word-level, the context-aware

architecture faces the limitation that compound words may cause the enlargement of search space. Code-aware search strategy aims to address the limitations. The CURE technique with three major novelties was presented [23]. The three novelties include a pre-trained programming language (PL) model-Generative Pre-trained Transformer (GPT), a new code-aware search strategy, and a sub word tokenization technique through the byte pair encoding. The PL model helps the NMT model in CURE, CoCoNuT, learns developer-like source code. The code-aware strategy aims to generate condign sequences for the NMT model. BPE tokenizes compound and rare words so that more correct fixes can be included and at the same time, shrink the size of search space. The CURE method observably enhances the compliable rates of the candidate patches while still guaranteeing the patches' precision.

Table 1. Automatic program repair.

Articles	Strategy	Models/Algorithms	Results
[20]	context-aware	AST	Defects4J: 21/25 (393bugs)
[21]	context-aware	Seq2Seq/LSTM	Defects4J: 12/19 (393bugs)
[22]	context-aware	Encoder-decoder/CNN	Defects4J: 44/85 (393 bugs) QuixBugs (Java): 13/20 (40 bugs) Codeflaws: 423/716 (3902 bugs) ManyBugs: 7/- (69bugs) QuixBugs (Python): 19/21 (40 bugs) BugAID: 3/- (12 bugs)
[23]	code-aware	CoNuT/BPE/GPT	Defects4J: 57/104 (393bugs) QuixBugs: 26/35 (40 bugs)

3.3. Simultaneous translation

Simultaneous translation is an important translation tool where the translating process begins before the speaker ends up speaking. More recently, there is great interest for researchers to study simultaneous machine translation, which is a special form of MT that the process of translating starts before the end of the input.

Generally, the translation process in simultaneous translation is divided into two stages: segmentation and translation. In this case, translation is done largely according to the segment, namely, it does not fit the context well. Therefore, attention-based neural translation is introduced and most research are based on it. Besides encoder and decoder, attentional mechanism is introduced in this model, which helps better fit the context. Traditional attention-based NT, however, cannot translate with sufficient delay in network scenarios. To solve this, the authors present a new attentional mechanism based on reinforced learning, which consists of two models, one to control translation quality and the other to control delay, allowing an NMT model to train the stopping criterion and partial translation model jointly [24].

Compared with conventional consecutive MT, simultaneous machine translation not only emphasizes the accuracy but also the time delay, so one of the most significant concerns in SMT is the problem of balancing latency and translation quality. To solve this problem, the authors proposed a novel decoding algorithm named simultaneous greedy decoding, which includes two adjustable parameters-s0 and δ, allowing the users to trade off translation delay and quality smoothly [25]. In another literature, the authors modified the architecture in the NMT decoder by proposing a tunable agent that can make the most appropriate segmentation decision according to AP constraints and user-defined BLEU loss [26]. A neural machine translation framework tailored for simultaneous translation

is proposed, where an agent can learn to decide when to start translating by interacting with a pre-trained environment [27]. Besides, to better balance translation latency and quality, the authors designed a beam-search method. The authors presented a new prediction action, which can be trained through reinforcement learning, to perfect quality and delay in simultaneous translation by predicting the future words in the input [28]. The authors introduced MILk (Monotonic Infinite Lookback) attention, which successfully realizes translation quality-delay balance, and then they proposed a simultaneous translation system that can learn an adaptive schedule [29]. In addition, an adaptive NMT method was presented to determine when to start to translate the observed words by introducing a special token '<wait>', which is generated when the NMT model starts to read the following input [30].

4. Discussion

The number of articles presented in this paper was 32 altogether. Among the articles, twenty-one of them are related to text NMT, which includes word-level NMT, short sentence-level NMT, character-level NMT, document level NMT and low-resource NMT, four articles are related to Automatic program repair (APR), while the other seven articles are related to the application of NMT in simultaneous translation. Figure 2 presents the contribution of articles according to the defined sub-categories.

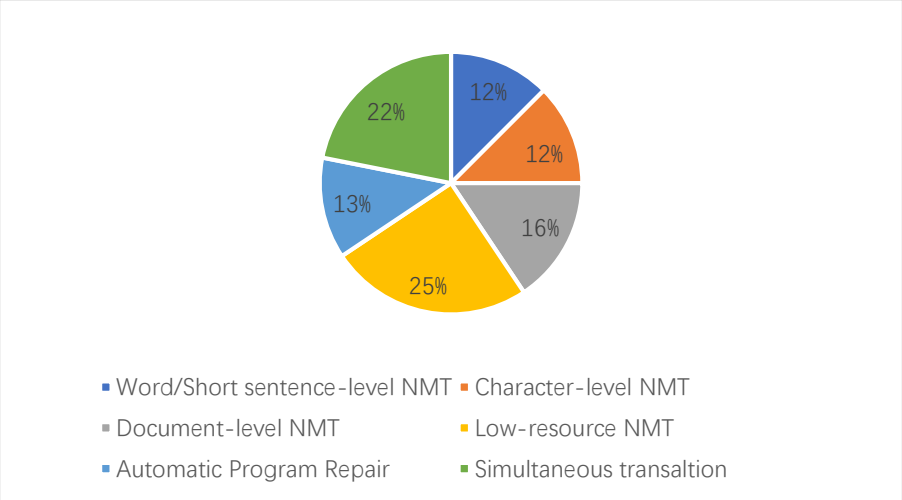

Figure 2. Pie chart presenting the articles according to the defined sub-categories.

From the analysis of the articles, it was found that compared with conventional SMT, NMT performed well in translating various types of text. In section 3.1.1, the original encoder-decoder architecture successfully finishes the translation task without unknown words. In the character-level NMT part, section 3.1.2, to settle the condition that accuracy declines rapidly when unknown words appear, character-level NMT was invented, which can predict the words that are not in the dictionary. Section 3.1.3 compares the document-independent sentence and document-level models and concludes that document-level models significantly improve the test performances. In section 3.1.4, the quality of low-resourcing translation significantly improved by using two techniques: Data augmentation techniques and NMT improvement. In the next part, this paper sorts the research on APR into two classifications: (i) APR using context-aware strategy (Section 3.2.1) and (ii)APR using code-aware strategy (Section 3.2.2) and introduces the features as well as typical models of each classification. In section 3.2.1, the introduced context-aware architecture works on the significant challenges NMT models face in APR. The models introduced to guarantee the precision of the generated patches despite the major challenge-the enlargement of search space due to the problem that the correct fixes are not likely to be included when the search space is relatively small. In section 3.2.2, the code-aware search strategy architecture enhances the compliable rate of the generated patches and makes efforts to handle the OOV problem.

Lastly, in 3.3, this work covered the research and recent progress on simultaneous machine translation, a particular form of NMT. The primary and mutual limitation that the NMT architectures face in the fields mentioned above is the OOV problem and the sentence sequencing problem. Previous techniques like BPE are presented to handle the problems. The performance of NMT models can be considerably boosted provided that new algorithms handle the problems above well.

5. Conclusion

With the emergence and development of neural network architectures, NMT models succeeded and rapidly became the primary trend of practical MT systems. In this paper, some critical applications of NMT were highlighted. In the beginning, this paper briefly introduced the encoder-decoder approaches as well as attentional mechanism, two architectures commonly used in NMT. Then starting with the survey of applications from text NMT, it has four sub-tasks to illustrate: (i) word/short sentence-level NMT, (ii) character-level NMT, (iii) document-level NMT, and (iv) low-resource NMT. In the next part, this paper sorts the research on APR into two classifications: (i) APR using context-aware strategy and (ii) APR using code-aware strategy. Lastly, this paper paid attention to simultaneous machine translation, a special form of NMT.

Even though NMT has made significant progress in recent years, some problems remain to be solved. For example, there is no perfect word segmentation algorithm so far to divide sentences into sequences, and the translating performances of various models are all restricted by OOV and UKN to varying degrees. For this scope, more efforts are expected to be made to research new decoding algorithms and attention mechanisms in the future. This essay summarizes the state-of-the-art research of different applications of NMT and comments on their development. It also notices the limitations of existing works and points out possible directions that could be meaningful in future studies.

References

[1] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate[J]. Computer Science, 2014.

[2] Cho K, Merrienboer B V, Bahdanau D, et al. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches[J]. Computer Science, 2014.

[3] Luong M T, Pham H, Manning C D. Effective Approaches to Attention-based Neural Machine Translation[J]. Computer Science, 2015.

[4] Chung J, Cho K, Bengio Y. A Character-level Decoder without Explicit Segmentation for Neural Machine Translation[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016.

[5] Luong M T, Manning C D. Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016.

[6] Costa-Jussà, Marta R, Fonollosa J. Character-based Neural Machine Translation[J]. arXiv preprint arXiv:1511.04586. 2016.

[7] Sennrich R, Haddow B, Birch A. Neural Machine Translation of Rare Words with Subword Units[J]. Computer Science, 2015.

[8] Wang L, Tu Z, Way A, et al. Exploiting Cross-Sentence Context for Neural Machine Translation[C]// Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017.

[9] Bawden R, Sennrich R, Birch A, et al. Evaluating Discourse Phenomena in Neural Machine Translation[C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). 2018.

[10] Rios A, Mascarell L,Rico Sennrich. Improving Word Sense Disambiguation in Neural Machine Translation with Sense Embeddings[C]// Conference on Machine Translation. 2017.

[11] Voita E, Sennrich R, Titov I. When a Good Translation is Wrong in Context: Context-Aware

Machine Translation Improves on Deixis, Ellipsis, and Lexical Cohesion[C]// 2019.
[12] Guo M, Shen Q, Yang Y, et al. Effective Parallel Corpus Mining using Bilingual Sentence Embeddings[C]// Proceedings of the Third Conference on Machine Translation: Research Papers. 2018.
[13] Lai G, Dai Z, Yang Y. Unsupervised Parallel Corpus Mining on Web Data[J]. 2020.
[14] Sennrich R, Haddow B, Birch A. Improving Neural Machine Translation Models with Monolingual Data[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016.
[15] Artetxe M, Labaka G, Casas N, et al. Do all roads lead to Rome? Understanding the role of initialization in iterative back-translation[J]. Knowledge-Based Systems, 2020.
[16] Ranathunga S, Lee E, Skenduli M P, et al. Neural Machine Translation for Low-Resource Languages: A Survey[J]. arXiv preprint arXiv:2106.15115. 2021.
[17] Xia Y, He D, Qin T, et al. Dual Learning for Machine Translation[C]// Advances in neural information processing systems. 2016.
[18] Pan S J, Qiang Y. A Survey on Transfer Learning[J]. 2009.
[19] Wang R, Tan X, Luo R, et al. A Survey on Low-Resource Neural Machine Translation[C]// 2021.
[20] Ming W, Chen J, Wu R, et al. Context-Aware Patch Generation for Better Automated Program Repair[C]// the 40th International Conference. IEEE Computer Society, 2018.
[21] Chen Z, Kommrusch S J, Tufano M, et al. SEQUENCER: Sequence-to-Sequence Learning for End-to-End Program Repair[J]. IEEE Transactions on Software Engineering, 2019, PP (99):1-1.
[22] Lutellier T, Pham H V, Pang L, et al. CoCoNuT: combining context-aware neural translation models using ensemble for program repair[C]// ISSTA '20: 29th ACM SIGSOFT International Symposium on Software Testing and Analysis. ACM, 2020.
[23] Jiang N, Letelier T, Tan L. CURE: Code-Aware Neural Machine Translation for Automatic Program Repair[C]// 2021.
[24] Lee Y H, Shin J H, Kim Y K. Simultaneous neural machine translation with a reinforced attention mechanism[J]. ETRI Journal. 2021.
[25] Cho K, Esipova M. Can neural machine translation do simultaneous translation? [J]. arXiv preprint arXiv:1610.00388. 2016.
[26] Dalvi F, Durrani N, Sajjad H, et al. Incremental Decoding and Training Methods for Simultaneous Translation in Neural Machine Translation[C]// North American Chapter of the Association of Computational Linguistics: Human Language Technologies. 2018.
[27] Gu J, Neubig G, Cho K, et al. Learning to Translate in Real-time with Neural Machine Translation[J]. arXiv preprint arXiv:1610.00388. 2017.
[28] Alinejad A, Siahbani M, Sarkar A. Prediction Improves Simultaneous Neural Machine Translation[C]// Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018.
[29] Arivazhagan N, Cherry C, Macherey W, et al. Monotonic Infinite Lookback Attention for Simultaneous Machine Translation[C]// Meeting of the Association for Computational Linguistics. 2019.
[30] Chousa K, Sudoh K, Nakamura S. Simultaneous Neural Machine Translation using Connectionist Temporal Classification[J]. arXiv preprint arXiv:1911.11933. 2019.