RNA Secondary Structure Prediction Using Neural Machine Translation

by

Luyao Zhang

A Project Presented to the
FACULTY OF THE OSU GRADUATE SCHOOL
OREGON STATE UNIVERSITY
In Partial Fulfillment of the
Requirements for the Degree
MASTER OF SCIENCE
(COMPUTER SCIENCE)

August 2016

Abstract

RNA secondary structure prediction maps a RNA sequence to its secondary structure (set of AU, CG, and GU pairs). It is an important problem in computational biology because such structures reveals crucial information about the RNAs function, which is useful in many applications ranging from noncoding RNA detection to folding dynamics simulation.

Traditionally, RNA structure prediction is often accomplished computationally by the cubic-time CKY parsing algorithm borrowed from computational linguistics, with the energy parameters either estimated physically or learned from data. With the advent of deep learning, we propose a brand-new way of looking at this problem, and cast it as a machine translation problem where the RNA sequence is the source language and the dot-parenthesis structure is the target language. Using a state-of-the-art open source neural machine translation package, we are able to build an RNA structure predictor without any hand-designed features.

Acknowledgments

I am deeply thankful to my advisor, Professor Liang Huang, whose inspiring guidance, generous support, and consistent trust have facilitated the success of this research. Professor Huang has not only offered me valuable guidance on my project, but also explored my interest in conducting impactful research.

Many thanks go to Prof. Fuxin Li and Prof. Yue Zhang for being my master committee members. They offered a lot of precious suggestions and feedbacks on my project which helped me to pursuit my goal. I would also like to thank Dezhong Deng and Wen Zhang, two members of our group, who helped me a lot on this project.

Finally, I would like to express my sincerest appreciation to my dear husband and parents for their unconditional love and support. Without their understanding, encouragement and sacrifice, I would not have all those achievements.

Contents

List of Figures

Chapter 1

Introduction

1.1 RNA Structure

RNA (Ribonucleic acid), along with DNA (deoxyribonucleic acid) and proteins, is one of the three major biological macromolecules that are essential for all known forms of life. It plays an important role in various biological processes, such as coding, decoding, regulation, and expression of genes. In recent years, researchers have found that RNA can also act as enzymes to speed chemical reactions.

RNA is a polymeric molecule assembled as a chain of nucleotides. Each nucleotide is made up of a base, a ribose sugar, and a phosphate. There are four types of nitrogenous bases, called cytosine (C), guanine (G), aenine (A), and uracil (U). The primary structure of RNA refers to the sequence of bases. Figure 1.1 shows the structure of RNA and DNA. Unlike DNA which is usually found in a paired double-stranded form in cells, RNA is a single-stranded molecule. However, owing to the hydrogen bonding between complementary bases on the same strand, most biologically active RNAs will partly pair and folded onto themselves[1] , forming the secondary structure of RNA.[2] In other words, the secondary structure of RNA can be represented as a list of bases which are paired in the molecule. Figure 1.2 shows an example of RNA secondary structure, which can be divided into helical regions composed of canonical base pairs (A-U, G-C, G-U), as well as single-stranded regions such as hairpin loops, internal loops, and junctions.

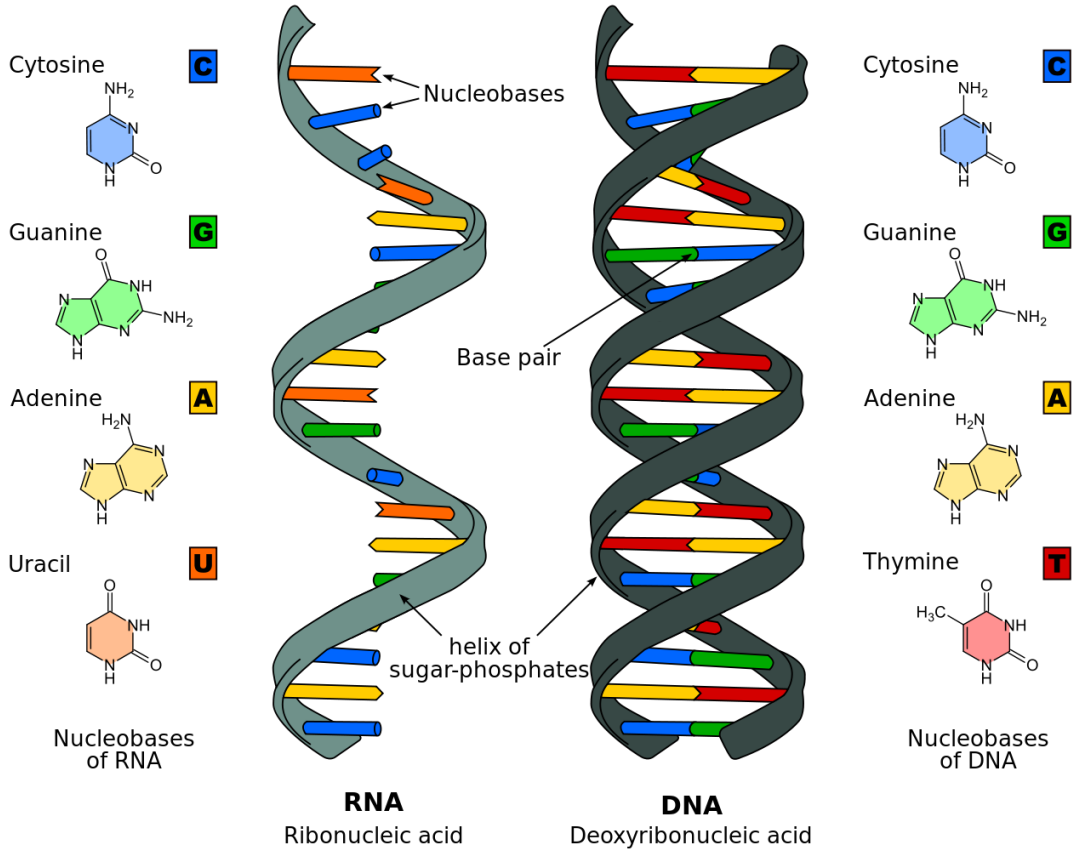

Figure 1.1: Structures of RNA and DNA

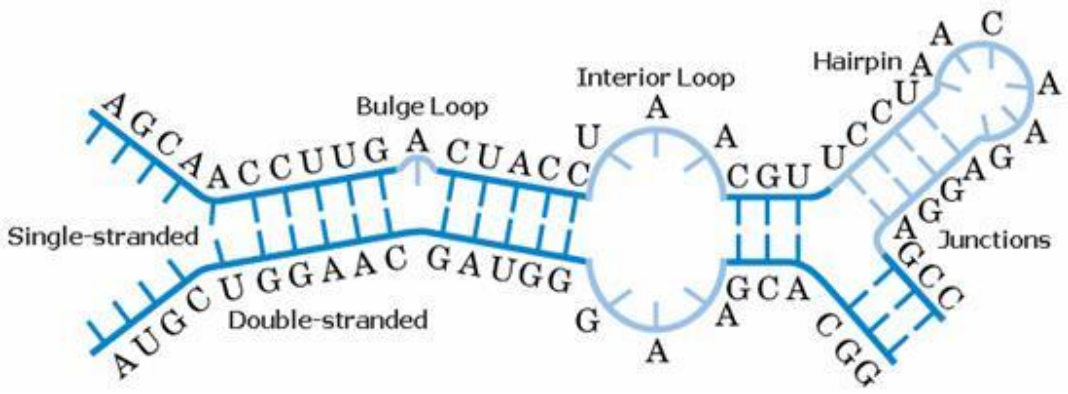

Figure 1.2: RNA molecule secondary structure motifs

single-strand regions, double-strand base pairs, hairpins, bulges, internal loops, junctions.

1.2 RNA Secondary Structure Prediction

RNA serves various roles in biological processes from modulating gene expression [3, 4] to catalyzing reactions[5, 6]. To understand RNAs' mechanism of action, the secondary structure must be known. The prediction of RNA structures has attracted increasing interest over the last decade. An efficient secondary structure prediction gives essential directions for experimental investigations.

There are two general approaches to predict the RNA secondary structure. One is to use experimental method such as NMR spectroscopy to identify the base pairing information.[7, 8] But these methods are difficult and expensive, which limit the high throughput applications. Therefore, computational method provides an alternative way to effectively predict the secondary structure of RNA.[9] In the following, several representative computational approaches are briefly reviewed.

1.2.1 Prediction Accuracy Evaluation

There are different ways to represent a RNA secondary structure. If we treat the RNA sequence as a string over the alphabet *A*, *C*, *G*, *U*, the primary structure of an RNA *B* can be written as $b_1,...,b_n$. Then, the secondary structure can be associated with each sequence *B* as a string *S* over the alphabet "(", "." ,")", where parentheses in *S* must be properly nested, and *B* and *S* must be compatible: If (s_i, s_j) are matching parentheses, then (b_i, b_j) must be a legal base-pair.[9] This notation is illustrated in figure 1.3(a).

Secondary structure prediction can be benchmarked for accuracy evaluation using

(a)

```
UGAGUUCUCGAUCUCUAAAAUCG
........((((.......))))
```

(b)

```
example: ACGUCAUGCA
gold:    (.(.).)().
test:    (()...)...
```

Figure 1.3: evaluation of RNA secondary structure prediction

(a) Representation of RNA sequence secondary structures with dot and parentheses. (b) Sample RNA sequence secondary structure prediction with gold and test results. Sensitivity (recall) = (red) / (red + blue) = 1/3, specificity (precision) = (red) / (red + purple) = 1/2, F-score = 2*(1/3)*(1/2) / (1/3 + 1/2) = 2/5.

sensitivity and specificity. Sensitivity is the percentage of known pairs predicted correctly (similar as the concept of recall in binary classification problem), and specificity is the percentage of correctly predict pairs in all predicted pairs (similar as precision). These two statistics are calculated as:

$$\text{sensitivity} = \frac{\text{number of correct base pairings}}{\text{number of true base pairings}}$$

$$\text{specificity} = \frac{\text{number of correct base pairings}}{\text{number of predicted base pairings}}.$$

F-score is equal to the harmonic mean of sensitivity and specificity, which is used to assess balanced prediction quality on sensitivity and specificity. Figure 1.3 (b) gives an example to show the concept.

1.2.2 Computational Approaches

Traditionally, the most successful computational techniques for single sequence secondary structure prediction are based on physics models of RNA structure. The method

relies on approximations of sequence-dependent stability for various motifs in RNA. The parameters used for computation typically come from empirical studies of RNA structural energetics. For a target RNA sequence, dynamic programming is used to identify candidate structures by free energy minimization.[10-12] The prediction accuracy of this method is generally high for short RNA sequences. With fewer than 700 nucleotides, the prediction accuracy can reach about 73%.[13] Despite the great success using thermodynamic rules, there are still some drawbacks that limit the improvement of prediction accuracy. The main reason is that the current algorithms are incomplete and could not characterize the whole complicated process of folding. For example, the effect of folding kinetics on RNA secondary structure was not taken into account.

1.2.3 Modeling Secondary Structure with SCFGs

Besides the computational method of RNA secondary structure prediction by free energy minimization, another type of computational approach is widely used, which is probabilistic modeling. Rather than conducting experiments to determine the thermodynamic parameters, probabilistic method uses model parameters that are directly derived from frequencies of different features learned from the set of known secondary structures.[14] Given an RNA sequence x, the goal is to output the most likely secondary structure y to maximize the conditional probability P(y|x). In most of these models, stochastic context free grammars (SCFGs) are used.[15]

To predict the structure of an RNA sequence, the basic idea is to construct a SCFG parse tree with production rules and corresponding probability parameters. The resulted most probable parsing tree determined by CKY parsing algorithm represents the most

likely secondary structure of RNA. For example, consider the following simple unambiguous transformation rules:[16]

S → aSu | uSa | cSg | gSc | gSu | uSg | aS | cS | gS | uS | ε

For a sequence x = agucu with secondary structure1 y = ((.)), the unique parse σ corresponding to y is S → aSu → agScu → aguScu → agucu. The SCFG models the joint probability of generating the parse σ and the sequence x as P(x, σ) = P(S → aSu) ·P(S → gSc) ·P (S → uS) ·P (S → ε).

1.2.4 CONTRAfold Model

A new RNA secondary structure prediction method is called CONTRAfold model, which is based on conditional random fields (CRF) and generalized upon SCFGs by using discriminative training and feature-rich scoring.[16] The features in CONTRAfold include base pairs, helix closing base pairs, hairpin lengths, helix lengths and so on, which closely mirror the features employed in traditional thermodynamic models. Figure 1.4 shows a list of those features.

$$
\begin{array}{lll}
\phi_{\text{hairpin base}} & \phi_{\text{hairpin length}}[\cdot] & \phi_{\text{helix base pair}}(\cdot,\cdot) \\
\phi_{\text{hairpin extend}} & \phi_{\text{helix change}}[\cdot] & \phi_{\text{helix closing}}(\cdot,\cdot) \\
\phi_{\text{helix extend}} & \phi_{\text{bulge length}}[\cdot] & \phi_{\text{single base pair stacking left}}((\cdot,\cdot),\cdot) \\
\phi_{\text{multi base}} & \phi_{\text{internal length}}[\cdot] & \phi_{\text{single base pair stacking right}}((\cdot,\cdot),\cdot) \\
\phi_{\text{multi unpaired}} & \phi_{\text{internal asymmetry}}[\cdot] & \phi_{\text{terminal mismatch}}((\cdot,\cdot),\cdot,\cdot) \\
\phi_{\text{multi paired}} & \phi_{\text{internal full}}[\cdot][\cdot] & \phi_{\text{helix stacking}}((\cdot,\cdot),(\cdot,\cdot))
\end{array}
$$

Figure 1.4: List of all potentials used in CONTRAfold model

CONTRAfold has a parameter γ to control the tradeoff between sensitivity and specificity. By adjusting the γ value, one can optimize for either higher sensitivity or higher

specificity with the use of maximum expected accuracy (MEA) algorithm for parsing. Figure 1.5 demonstrates the accuracy performance comparison of CONTRAfold method with other methods.[16] When γ is 6, sensitivity is 0.7377 and specificity is 0.6686. The prediction accuracy (f-score) reaches about 70%, which is higher than other probabilistic methods for modeling RNA structures.

Although these results are already very good, there is still space for improvement. Since all the previous works are based on the features from thermodynamic models which are not complete to represent the whole RNA folding process, we are curious whether there is a method to overcome this limitation. In recent years, deep learning achieves great success in many research fields and real applications. It can extract better features automatically. In this study, we will apply deep learning to learn the structures automatically.

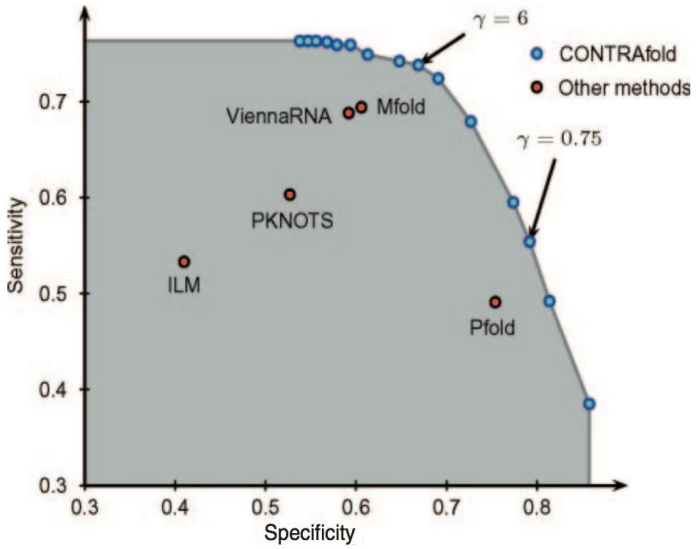

Figure 1.5: Comparison of sensitivity and specificity for several RNA secondary structure prediction methods

1.3 Neural Machine Translation

Neural machine translation (NMT) is a new approach to statistical machine translation.[17, 18] Most of the proposed neural machine translation models often consist of an encoder and a decoder.[19, 20] As figure 1.6 shows, an encoder neural network reads and encodes a variable-length source sentence into a fixed-length vector. A decoder then outputs a translation from the encoded vector. The whole system is jointly trained to maximize the probability of a correct translation given a source sentence.

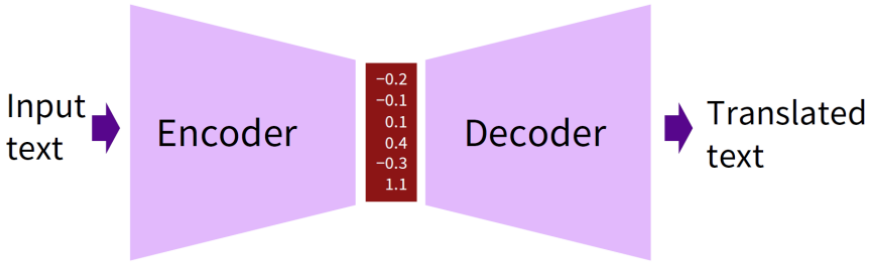

Figure 1.6: neural machine translation encoder-decoder architecture

The encoders and decoders are often realized by recurrent neural network (RNN). Figure 1.7 is an example of two layers of RNN for English-to-French translation from a source sentence “I am a student” into a target sentence “Je suis étudiant”. [21, 22] Here, “_” marks the end of a sentence.

The original neural machine translation model with fixed-length vector performs relatively well on short sentences without unknown words, but its performance degrades rapidly as the length of the sentence and the number of unknown words increase. This is mainly due to the word order divergence between different languages. Therefore, the attention mechanism which would be discussed in the following is introduced into NMT.

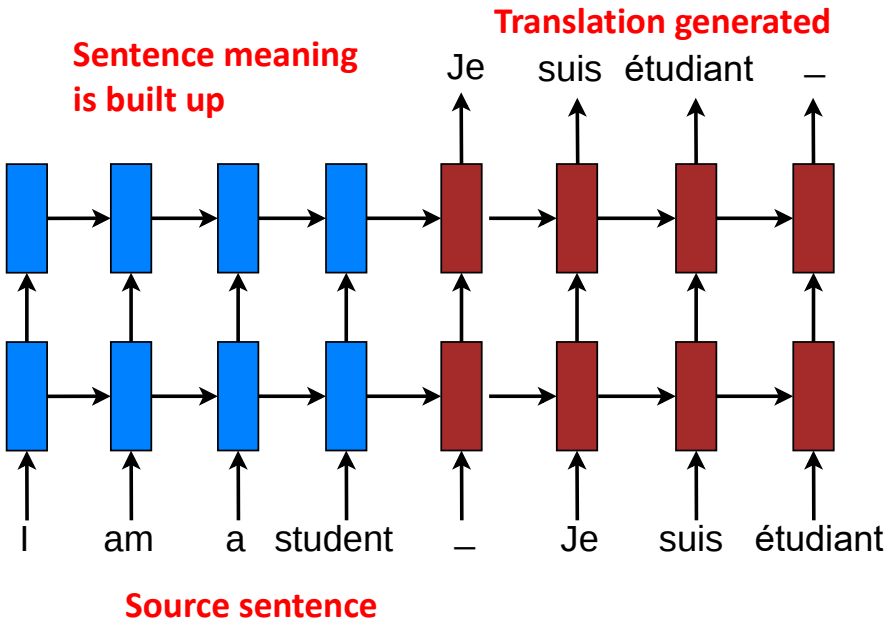

Figure 1.7: Neural machine translation: example of a deep recurrent architecture

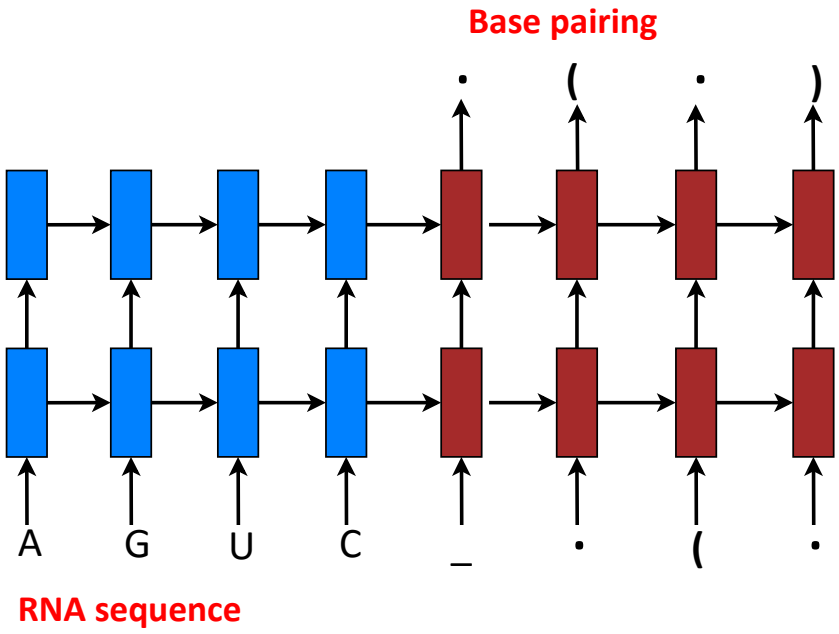

Figure 1.8: RNA folding prediction as a neural machine translation problem

1.3.1 RNA folding as Neural Machine Translation

In our task, we want to predict the RNA secondary structure given the RNA primary structure of base sequence. This is similar to the machine translation problem where the source sentence is a RNA sequence, and the translated sentence is the corresponding RNA base pairing. Therefore, the RNA secondary structure prediction is cast as a neural machine translation task and can be solved by this model using deep learning. Figure 1.8 illustrated this concept. The RNN models trained on RNA sequences with known structures can be applied to calculate the most likely base pairing for an unknown structure sequence without using any hand-designing features.

1.3.2 Attention Mechanism

Attention mechanism in neural networks is previously used for tasks like image captioning and recognition.[23-25] With the attention mechanism, the image is firstly divided into several parts, and a representation for each part is computed by a Convolutional Neural Network (CNN). When the RNN is trying to generate a new word or label from the image, the attention mechanism will focus on the relevant part of that image, and the decoder will only use the specific part of information.

To address the problem in previous neural machine translation model, the attention mechanism is applied to allow a model to automatically (soft-)search for parts of a source sentence that are relevant to predicting a target word.[22] This is also called soft-alignments. In figure 1.9, to generate the target word from the source sentence, a global context vector is computed as the weighted average according to an align weights vector based on the current target state and all the source state.

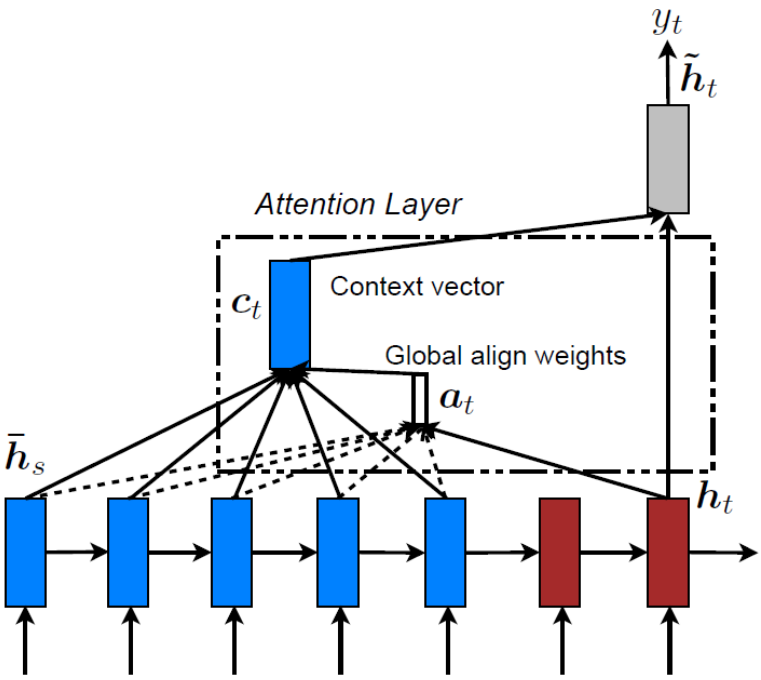

Figure 1.9: NMT model with attention mechanism

1.3.3 Encoding and Decoding Strategies

In this study, we adopt the encoder-decoder structure proposed in Bahdanau et al. 2014 work. [17] The encoder is a bidirectional RNN with soft alignment as figure 1.10 shows. The forward and backward RNN contain the information of both the preceding words and the following words.

In the decoder, we use beam search algorithm instead of greedy. In figure 1.11, beam-search finds a translation that maximizes the conditional probability given by a specific model. At each time step of the decoder, we keep the top beam-width translation candidates with the decreasing probability.

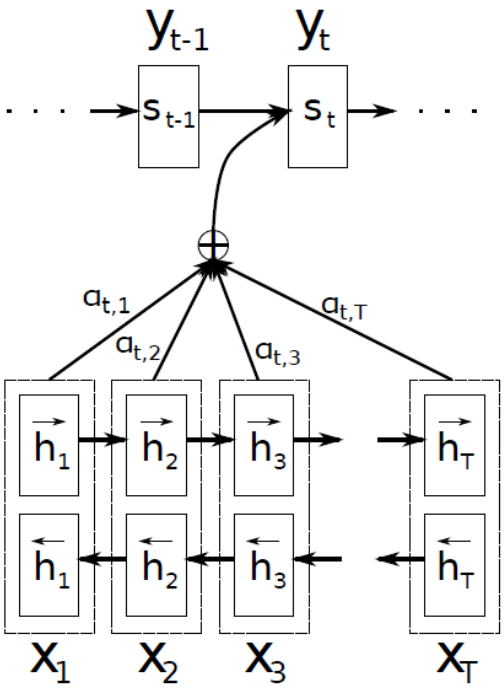

Figure 1.10: The graphical illustration of using bidirectional RNN for encoding

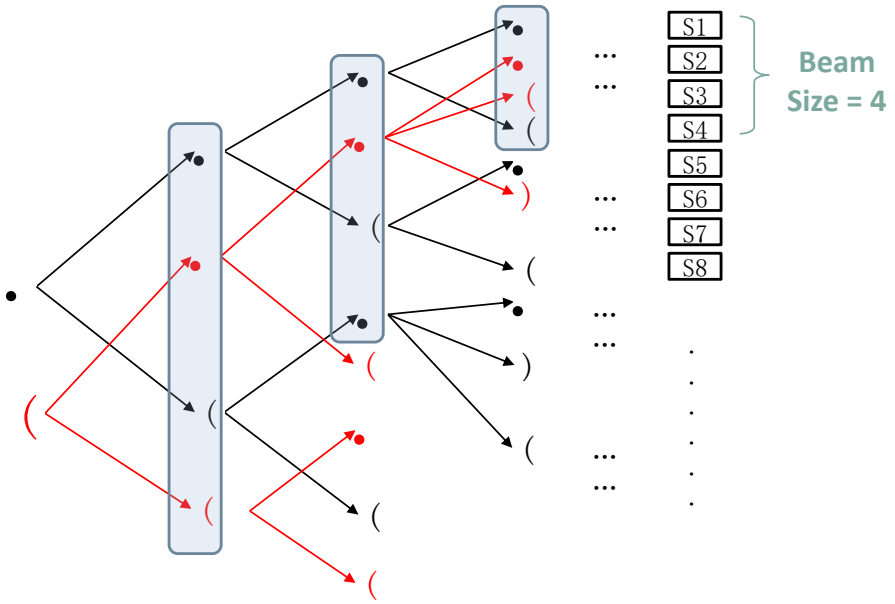

Figure 1.11: Beam search algorithm to select the candidate RNA structures Here, beam size is 4. Candidate structures are sorted by probability scores.

Chapter 2

Experiment Setup

2.1 Dataset and Code Source

The RNA dataset we used is from Rfam database and has 151 files. Each file contains one RNA base sequence and corresponding secondary structure. Among these 151 sequences, the average nucleic acid length is 136, and the maximum length is 568. The data is pre-selected to ensure that the RNA secondary structures like pseudo-knots are not present in those sequences. This one set of data is randomly selected and divided into three sets: 80% for training, 10% for validation, and 10% for testing.

We built our model based on the state-of-the-art neural machine translation open source package.[26]

2.2 Modification on Encoder and Decoder

Although the task of machine translation of a sentence from one language to another is similar to the RNA secondary structure prediction, there are still some differences that require modifications on the training and testing process. One major difference is that the input and output sentence length in machine translation has not to be the same. In most cases, the source and target length are different. For RNA structure prediction, the input is a base sequence and the output is the dot-parenthesis base pairings, which must hold

the same length. In the encoder part, for the original design, there is an <EOS> tag to mark the end of input source sentence. In our code, we developed a modified version of encoder which removed the <EOS> tag in the input. In the decoder part, we force the output length to be the same as input sequence.

On the other hand, RNA base pairing rules also need to be taken into account since in our cases, only C-G, A-U, G-U pairs are allowed. To make sure the output RNA secondary structures will follow these rules, in the decoder, we add a stack to keep track of the left parenthesizes and their corresponding base types. Whenever the model is going to predict a right parenthesis, it will locate the left pairing parenthesis and check whether the left and right bases obey the three pairing rules. If not, the prediction of that right parenthesis will not be allowed. Also, if there is no left parenthesis in the stack, the right parenthesis prediction is also forbidden. In this way, the output secondary structures are guaranteed to follow the base pairing rules.

2.3 Three Encoder Decoder Systems

2.3.1 System No.1: Naïve Model

The first system is a baseline for performance comparison. In this naïve model, the <EOS> tag is kept, and the generation of <EOS> marks the completion of input. The maximum output length allowed is set to be 600. There is no stack in the decoder to control the output pairing as well. This is the original model adapted from the NMT package.

2.3.2 System No.2: Add Length Control

The second system is a modified version based on the Naïve model. It removed the <EOS> tag in the encoder. We force the translated output dot-parenthesis sequence length to be the same as the input RNA bases length. However, there is no stack in the decoder to enforce the pairing rule.

2.3.3 System No.3: Add Pairing Rule Control

The third system takes one step further from the second one. On top of system 2, we add the stack in the decoder to ensure the base pairing rules are followed. Figure 2.1 shows the relationship of these three systems.

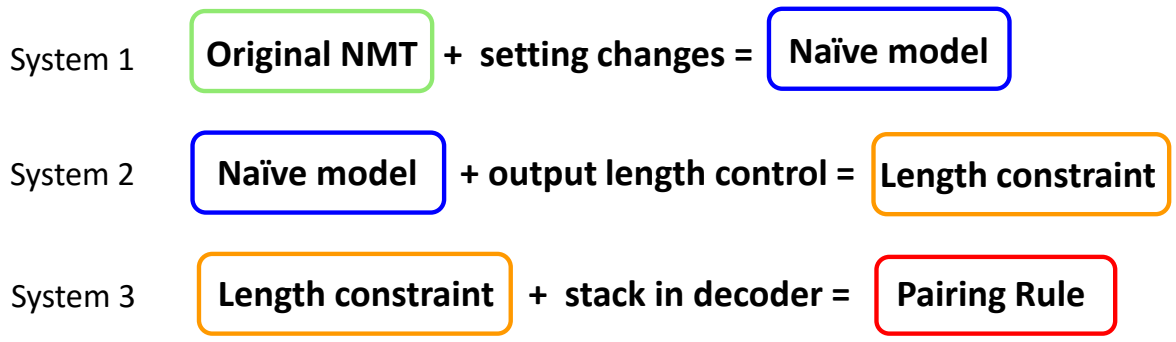

Figure 2.1: Three systems

Chapter 3

Results and Discussions

3.1 Evaluation Metrics

Beside specificity, sensitivity and f-score, for this RNA secondary structure prediction task, we further introduced four performance evaluation metrics: position accuracy, left parenthesis recall, right parenthesis recall and dot recall. The definitions of these parameters are shown below in figure 3.1. These parameters only consider the true label and the predicted label for one base and don't compare the whole pairs. The position accuracy is a composite ratio of left, right and dot recall. This number must stays between these three recalls. Since RNA sequences have different lengths, all the correct predicted A (A= "(", ".", ")") labels in each sequence are added together and be divided by the sum of the true A labels in those sequences during the evaluation.

$$\text{Position accuracy} = \frac{number\ of\ correctly\ predicted\ positions}{length\ of\ the\ sequence}$$

$$\text{Left (recall} = \frac{number\ of\ correctly\ predicted\ "("\ labels}{number\ of\ true\ "("\ labels}$$

$$\text{Right) recall} = \frac{number\ of\ correctly\ predicted\ ")"\ labels}{number\ of\ true\ ")"\ labels}$$

$$\text{Dot . recall} = \frac{number\ of\ correctly\ predicted\ "."\ labels}{number\ of\ true\ "."\ labels}$$

Figure 3.1: Definition of evaluation metrics

3.2 Performance of Three Systems

The preliminary results show that the naïve model has very low sensitivity and specificity, and the f-score is below 2%. This is reasonable since the first system doesn't have any constraint on encoding and decoding. We will briefly go over the results from system 2 and put most focus on system 3.

Figure 3.2 is the performance of system 2 models tested on validation set with increasing training epochs and sparse sampling. In figure (a), the M-shape curves indicate that in the beginning of training, the model does not learn the features and the performance is poor. When it starts to learn the features, performance rises quickly and reaches the optimal value. With more training epochs, the model is over fit and performance degrades. During all training epochs, sensitivity is always higher than specificity. This means there are a lot of wrong predicted pairs. In figure (b), dot recall > left recall > right recall, and the position accuracy sits in between. With more training epochs, dot recall drops while left and right recall rise, which means that models output mostly dot sequences at the beginning and later on starts to output parenthesis. The right recall is lower than the left recall because the output sequences usually contain more left parenthesis than the right ones. Comparing the results of (a) and (b), we can see that sensitivity and specificity are much lower than position accuracy and dot-parenthesis recalls. While sensitivity and specificity are more task-based evaluation metrics, the parenthesis-dot recalls are more model-based evaluation metrics. Both specificity and sensitivity require a correct prediction of a pair. But position recall only looks at one position at a time and check whether it is correct.

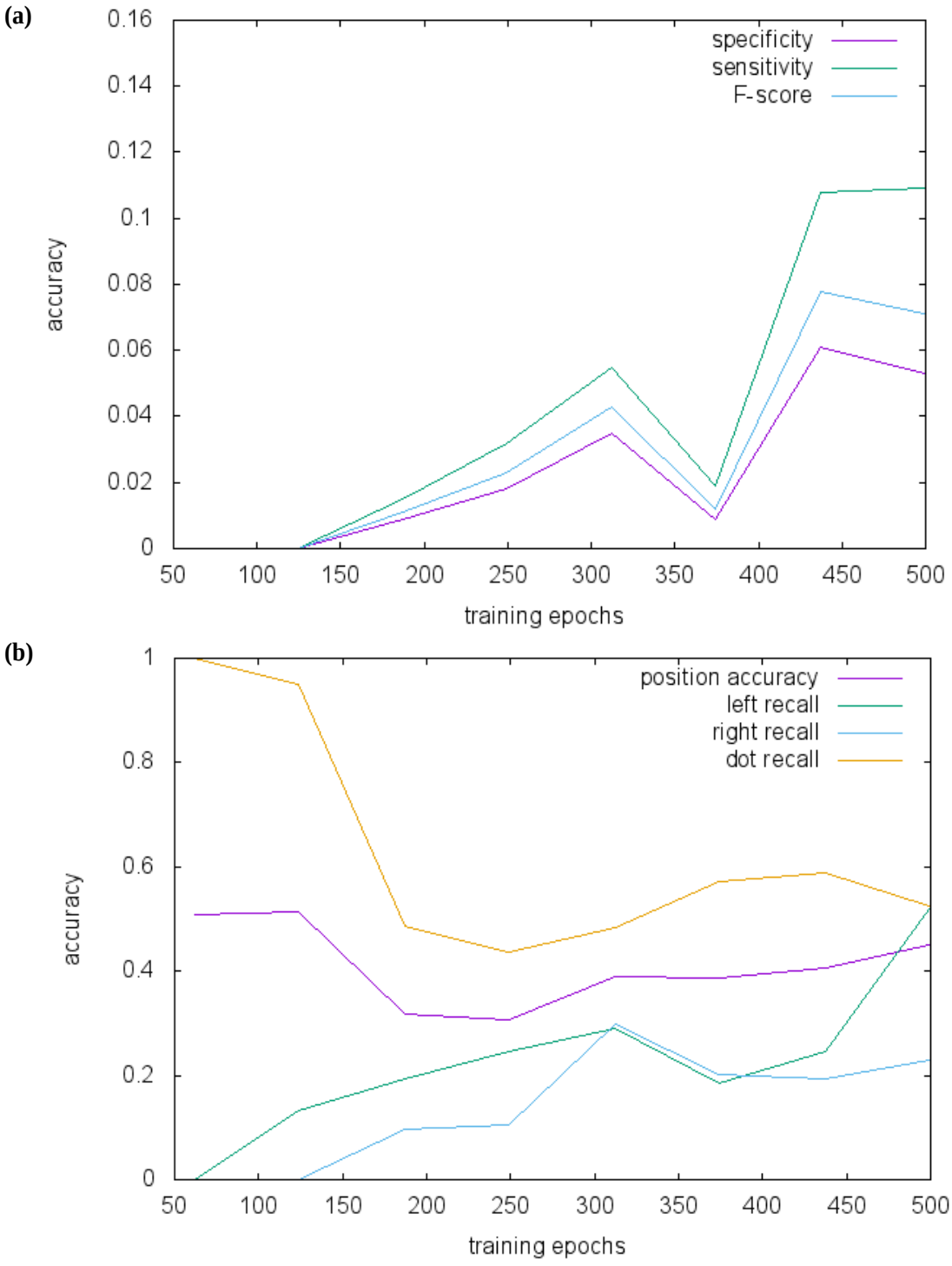

Figure 3.2: Accuracy change on training epochs tested on validation set with system 2

It is much more difficult to predict a correct pair, since any wrong prediction between the left and right parenthesis may ruin the sequence and cause wrong pairing for the following bases. In other words, the pair prediction process is very sensitive and vulnerable to wrong prediction. Therefore, the position accuracy is much higher than sensitivity and specificity.

Figure 3.3 is the position accuracy and recalls of dot-parenthesis on validation set with more training epochs. The four curves fluctuate in a large range which indicates that the model is not converging through the training epochs. This is a sign that the training process doesn't learn the features well.

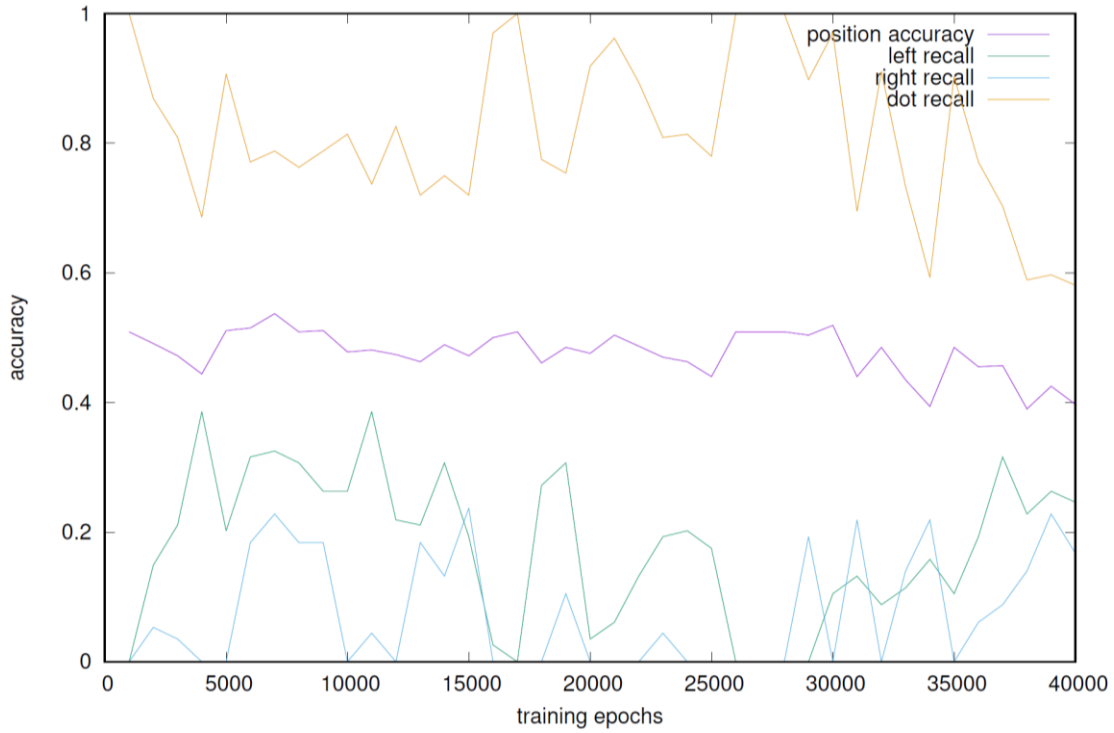

Figure 3.3: Accuracy change on more training epochs tested on validation set with system 2

Figure 3.4 is the prediction accuracy of system 3 on training set with increasing training epochs. From the results in panel (a), we see that sensitivity is still higher than specificity and both parameters increase slowly with training epochs. The f-score is around 0.1 for training epochs around 1200. In panel (b), the four curve shapes are similar as system 2. But in system 3, right recall is much lower than that in system 2. It is expected since system 3 adds the stack for pairing rule control. A large percent of right parenthesis prediction is greatly suppressed by the controlling rule in the decoder. Therefore, the right recall only reaches less than half of the previous value.

Figure 3.5 is the performance of trained models on validation set. In panel (a), sensitivity, specificity and f-score demonstrate a reversed V shape and the peak value is around 750 epochs. After that, the performance starts to decrease. This indicates that the models might have been over fitted after 750 epochs. In panel (b), the shape of four curves is similar as in figure 3.4 with a slight drop.

Comparing the results from figure 3.2 and 3.5, which are both tested on validation set, we can tell that performance of system 3 is better than system 2 as expected.

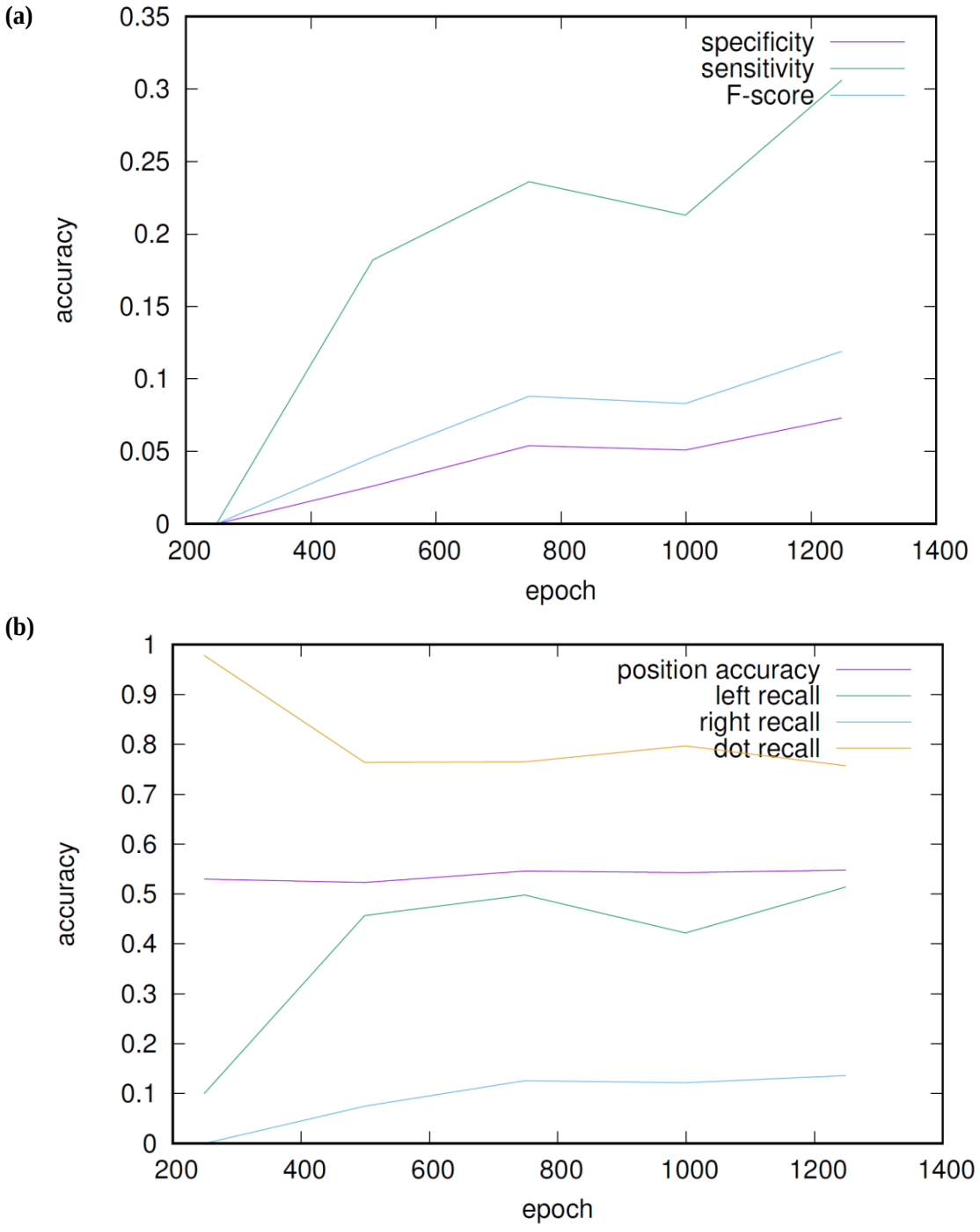

Figure 3.4: Accuracy change on training epochs tested on training set with system 3

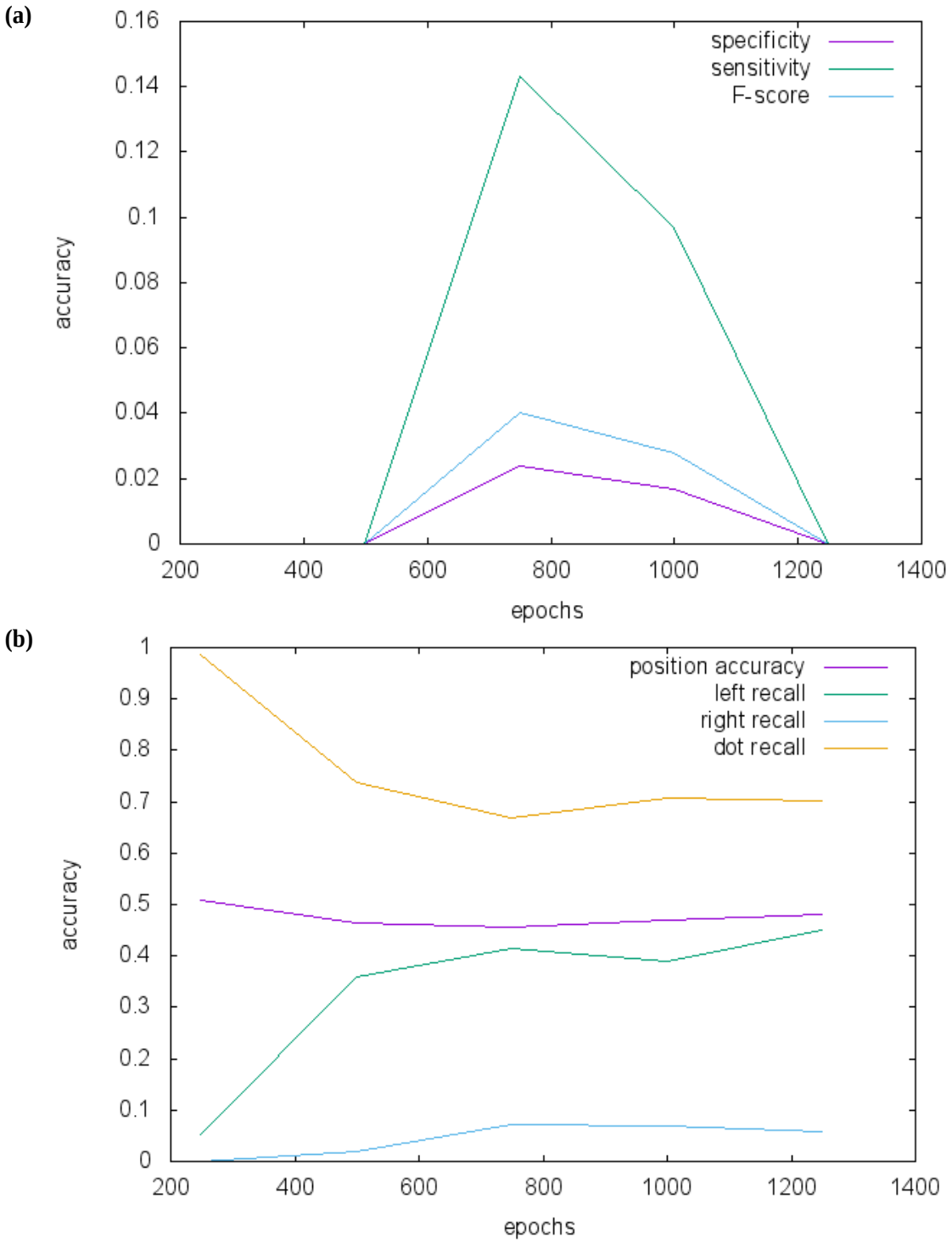

Figure 3.5: Accuracy change on training epochs tested on validation set with system 3

From the comparison of results of three systems, we can see that adding the length and pairing rule control improve the prediction performance. The best results of sensitivity and specificity is achieved by system 3. Comparing our model with the CONTRAfold model in table 3.1, the accuracy is lower than expected. The possible reasons will be discussed in section 3.4.

Table 3.1: Prediction Accuracy Comparison

Methods	Sensitivity	Specificity	F-score
CONTRAfold	0.738	0.669	0.701
Our Model	0.061	0.127	0.083

3.3 Beam Search Size Tuning

Based on the previous results on different systems, we used the third system to test the effect of tuning beam search size. The prediction accuracy with different beam widths (1, 2, 4, 8, 16) is shown in figure 3.6. The x axis is the beam size and the y axis is the accuracy. From the two images, we can see that there are dips in front part of the curves, indicating that increasing beam search size does not guarantee accuracy improvement at the beginning of beam width tuning. A possible reason is that in the first few steps of prediction, the score of the top candidates are very close and one good candidate may turn out to be bad in the future, which causes fluctuation in the curve. When beam search size increased to 8, the performance is almost stable and reaches the optimal value. Further increasing beam width doesn't help to improve the prediction accuracy.

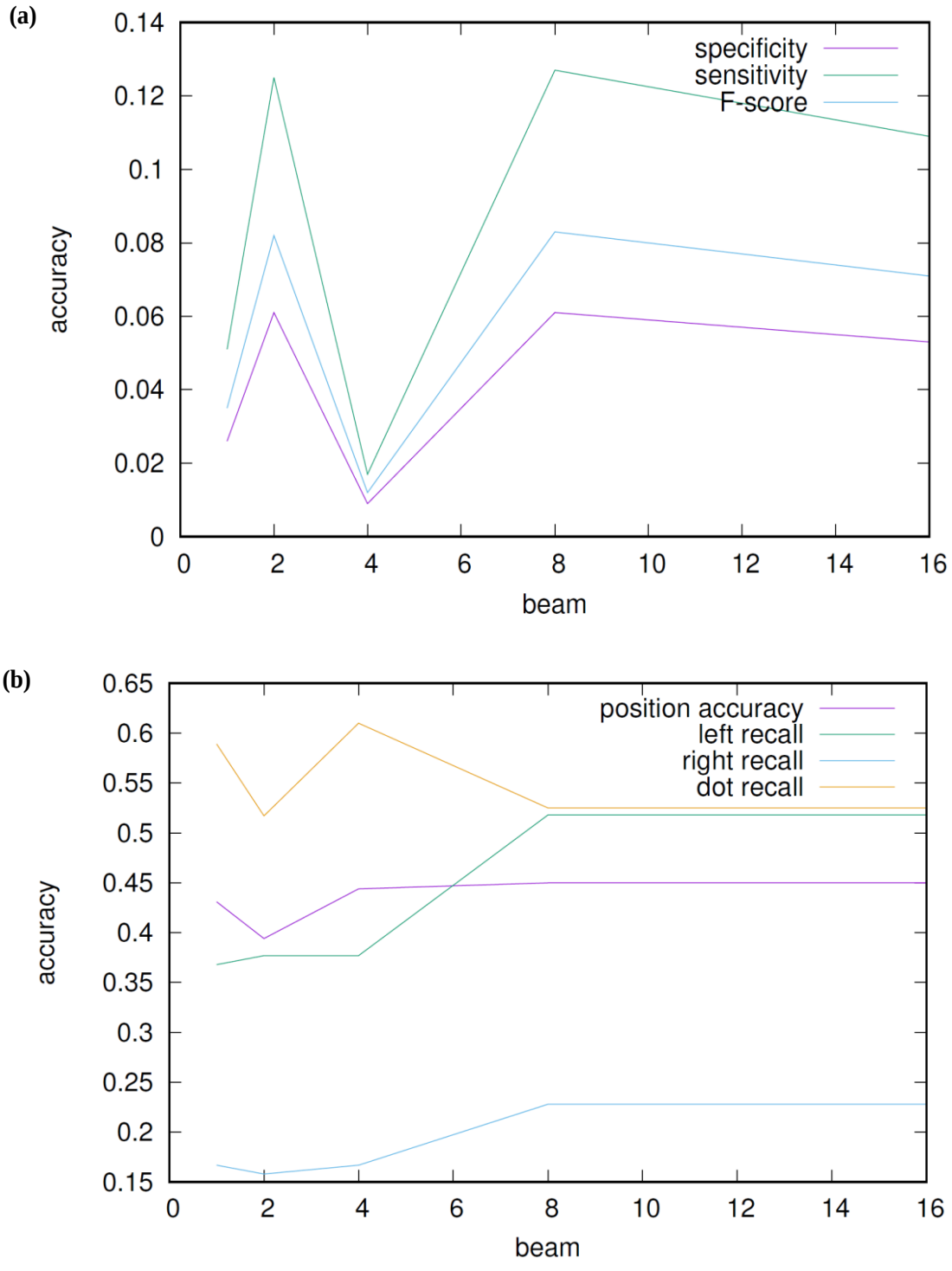

Figure 3.6: Accuracy change on training epochs tested on validation set with system 3

3.4 Results Analysis

This work is the first work that applies deep learning with neural machine translation model for RNA structure prediction. Although the performance is not as good as the previous traditional computation method, it gives us some insights in studying. There are several facts that limit the prediction accuracy of this method. The first one is that the dataset used in this study is relatively small compared to other tasks using deep learning. With this small data size, it is tricky to train a fine model.

Another important reason is that the machine translation model is not specified for this task and there are some key differences. For machine translation, the vocabulary size is very large, and each word has some meanings, where word embedding is used to represent it in a vector, while in this task, the vocabulary size is only three: the left parenthesis, the right parenthesis and the dot. This is too small and does not contain much information. On the other hand, for machine translation from one language to another, typically the input sentences are not too long (with less than 50-60 words). However, in our task, the input sequences have an average length of 136, which is a very long sentence for machine translation. It adds the difficulty for the model training. Our test on another dataset with even longer RNA sequences is not successful and very time consuming, implying that long input sequences with small vocabulary size is not suitable in this model.

Furthermore, the architecture of our system separates encoding and decoding process. Training and testing have different goals and behaviors: training is local and greedy, which optimizes local probability, and assuming correct prediction history; testing wants to improve the f-score, but produces mistakes along the way. Therefore, a good model in

the training process may not be a good one in the testing. From the accuracy results, it seems that the model trained does not learn the pairing behavior of RNA bases and interprets it during prediction.

3.5 Conclusions

In conclusion, we conducted the RNA sequence secondary structure prediction using neural machine translation method for the first time. Our NMT model is based on a bidirectional RNN encoder and decoder system. Unlike the previous methods which need pre-designed features in the model, our approaches apply deep learning and do the feature extraction automatically. We add length control and pairing rule constraint to the original model to improve the prediction accuracy and the performance from three systems are compared. The results show that system with length control and pairing rule control is better than the other two.

Bibliography

[1] Tinoco, I.; Bustamante, C., How RNA folds. *Journal of molecular biology* 1999, 293, 271-281.

[2] Higgs, P. G., RNA secondary structure: physical and computational aspects. *Quarterly reviews of biophysics* 2000, 33, 199-253.

[3] Wu, L.; Belasco, J. G., Let me count the ways: mechanisms of gene regulation by miRNAs and siRNAs. *Molecular cell* 2008, 29, 1-7.

[4] Storz, G.; Gottesman, S., 20 Versatile Roles of Small RNA Regulators in Bacteria. *Cold Spring Harbor Monograph Archive* 2006, 43, 567-594.

[5] Rodnina, M. V.; Beringer, M.; Wintermeyer, W., How ribosomes make peptide bonds. *Trends in biochemical sciences* 2007, 32, 20-26.

[6] Doudna, J. A.; Cech, T. R., The chemical repertoire of natural ribozymes. *Nature* 2002, 418, 222-228.

[7] Fürtig, B.; Richter, C.; Wöhnert, J.; Schwalbe, H., NMR spectroscopy of RNA. *Chembiochem* 2003, 4, 936-962.

[8] Latham, M. P.; Brown, D. J.; McCallum, S. A.; Pardi, A., NMR methods for studying the structure and dynamics of RNA. *Chembiochem* 2005, 6, 1492-1505.

[9] Gardner, P. P.; Giegerich, R., A comprehensive comparison of comparative RNA structure prediction approaches. *BMC bioinformatics* 2004, 5, 1.

[10] Hofacker, I. L.; Fontana, W.; Stadler, P. F.; Bonhoeffer, L. S.; Tacker, M.; Schuster, P., Fast folding and comparison of RNA secondary structures. *Monatshefte für Chemie/Chemical Monthly* 1994, 125, 167-188.

[11] Zuker, M., Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic acids research* 2003, 31, 3406-3415.

[12] Ying, X.; Luo, H.; Luo, J.; Li, W., RDfolder: a web server for prediction of RNA secondary structure. *Nucleic acids research* 2004, 32, W150-W153.

[13] Mathews, D. H.; Disney, M. D.; Childs, J. L.; Schroeder, S. J.; Zuker, M.; Turner, D. H., Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences of the United States of America* 2004, 101, 7287-7292.

[14] Dowell, R. D.; Eddy, S. R., Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC bioinformatics* 2004, 5, 1.

[15] Knudsen, B.; Hein, J., RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics* 1999, 15, 446-454.

[16] Do, C. B.; Woods, D. A.; Batzoglou, S., CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics* 2006, 22, e90-e98.

[17] Bahdanau, D.; Cho, K.; Bengio, Y., Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* 2014.

[18] Luong, M.-T.; Sutskever, I.; Le, Q. V.; Vinyals, O.; Zaremba, W., Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206* 2014.

[19] Cho, K.; Van Merri ënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y., Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* 2014.

[20] Cho, K.; Van Merri ënboer, B.; Bahdanau, D.; Bengio, Y., On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* 2014.

[21] Luong, M.-T.; Manning, C. D. In *Stanford Neural Machine Translation Systems for Spoken Language Domains*, Proceedings of the International Workshop on Spoken Language Translation, 2015.

[22] Luong, M.-T.; Pham, H.; Manning, C. D., Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* 2015.

[23] Mnih, V.; Heess, N.; Graves, A. In *Recurrent models of visual attention*, Advances in Neural Information Processing Systems, 2014; pp 2204-2212.

[24] Ba, J.; Mnih, V.; Kavukcuoglu, K., Multiple object recognition with visual attention. *arXiv preprint arXiv:1412.7755* 2014.

[25] Gregor, K.; Danihelka, I.; Graves, A.; Rezende, D. J.; Wierstra, D., DRAW: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623* 2015.

[26] arctic-nmt, https://github.com/laulysta/nmt/tree/master/nmt. 2015.