Exploration on Generating Traditional Chinese Medicine Prescriptions from Symptoms with an End-to-End Approach

Wei Li[1] and Zheng Yang[2]

[1] MOE Key Lab of Computational Linguistics, School of EECS, Peking University
liweitj47@pku.edu.cn
[2] School of Traditional Chinese Medicine, Beijing Univeristy of Chinese Medicine
yangzheng@bucm.edu.cn

Abstract. Traditional Chinese Medicine (TCM) is an influential form of medical treatment in China and surrounding areas. In this paper, we propose a TCM prescription generation task that aims to automatically generate a herbal medicine prescription based on textual symptom descriptions. Sequence-to-sequence (seq2seq) model has been successful in dealing with sequence generation tasks. We explore a potential end-to-end solution to the TCM prescription generation task using seq2seq models. However, experiments show that directly applying seq2seq model leads to unfruitful results due to the repetition problem. To solve the problem, we propose a novel decoder with coverage mechanism and a soft loss function. The experimental results demonstrate the effectiveness of the proposed approach. Judged by professors who excel in TCM, the generated prescriptions are rated 7.3 out of 10, which means that the model can indeed help with the prescribing procedure in real life.

Keywords: Traditional Chinese Medicine · Prescription Generation · End-to-End Method.

1 Introduction

Traditional Chinese Medicine (TCM) is one of the most important forms of medical treatment in China and the surrounding areas. TCM has accumulated large quantities of documentation and therapy records in the long history of development. Prescriptions consisting of herbal medication are the most important form of TCM treatment. TCM practitioners prescribe according to a patient's symptoms. The patient takes the decoction made out of the herbal medication in the prescription. A complete prescription includes the composition of herbs, the proportion of herbs, the preparation method and the doses of the decoction. In this work, we focus on the composition part of the prescription, which is the most important.

During the long history of TCM, there has been a number of therapy records or treatment guidelines in the TCM classics composed by outstanding TCM practitioners. In real life, TCM practitioners often take these classical records

Table 1. An example of a TCM symptom-prescription pair. We only concern about the composition of the prescription in this task.

Name	麻黄汤 (Mahuang decoction)
Symptoms	外感风寒表实证。恶寒发热，头身疼痛，无汗自喘，舌苔薄白，脉浮紧。
Translation	Affection of exogenous wind-cold; aversion to cold, fever; headache and body pain; adiapneustia and pant; thin and white tongue coating, floating and tense pulse
Prescription	麻黄、桂枝、杏仁、甘草
Translation	Mahuang (ephedra), Guizhi (cassia twig), Xingren (almond), Gancao (glycyrrhiza)

for reference when prescribing for the patient, which inspires us to design a model that can automatically generate prescriptions by learning from these classics. It also needs to be noted that due to the issues in actual practice, the objective of this work is to generate candidate prescriptions to facilitate the prescribing procedure instead of completely substituting the human practitioners.

An example of TCM prescription is shown in Table 1. The herbs in the prescription are organized in a weak order. By "weak order", we mean that the effect of the herbs are not influenced by the order. However, the order of the herbs reflects the way of thinking when constructing the prescription. Therefore, the herbs are connected to each other, and the most important ones are usually listed first. Due to the lack of digitalization and formalization, TCM has not attracted sufficient attention in the artificial intelligence community. To facilitate the studies on automatic TCM prescription generation, we collect and clean a large number of prescriptions as well as their corresponding symptom descriptions from the Internet.[3]

Inspired by the great success of natural language generation tasks like neural machine translation (NMT) [1, 2, 7], abstractive summarization [6], generative question answering [12], and neural dialogue response generation [5, 4], we propose to adopt the end-to-end paradigm, mainly the sequence to sequence model, to tackle the task of generating TCM prescriptions based on textual symptom descriptions.

The sequence to sequence model (seq2seq) consists of an encoder that encodes the input sequence and a decoder that generates the output sequence. The success in the language generation tasks indicates that the seq2seq model can learn the semantic relation between the output sequence and the input sequence well. It is also a desirable characteristic for generating prescriptions according to the textual symptom description.

In the TCM prescription generation task, the textual symptom descriptions can be seen as the question in the question answer setting and the aim of the task is to produce a set of TCM herbs that form a prescription as the answer to

[3] The resources will be published online.

the question. However, the set of herbs is different from the textual answers. One difference that is most evident is that there should not be any duplication of herbs in the prescription. However, the basic seq2seq model sometimes produces the same herb tokens repeatedly when applied to the TCM prescription generation task. This phenomenon can hurt the performance of recall rate even after we apply a post-process to eliminate repetitions. Because in a limited length of the prescription , the model would produce the same token over and over again, rather than real and novel ones. Furthermore, the basic seq2seq assumes a strict order between generated tokens, but in reality, we should not severely punish the model when it predicts the correct tokens in the wrong order.

In this paper, we propose the task to automatically generate TCM prescriptions based on textual symptoms. We propose a soft seq2seq model with coverage mechanism and a soft loss function. The coverage mechanism is designed to make the model aware of the herbs that have already been generated while the soft loss function is to relieve the side effect of strict order assumption. In the experiment results, our proposed model beats all the baselines. The results also show the practicability of the generated prescriptions in professional evaluations.

The main contributions of this paper lie in the following three folds:

- We propose a TCM prescription generation task and collect a large quantity of TCM prescription data including symptom descriptions. It is the first time that this task has been considered to our knowledge.
- We propose to enhance the basic seq2seq model with cover mechanism and soft loss function to guide the model to generate more fruitful results, thus increasing the recall rate.
- In the experiments, the professional human evaluation score reaches 7.3 (out of 10), which shows that our model can indeed help the TCM practitioners to prescribe in real life. Our final model also increases the F1 score and the recall rate in automatic evaluation by a substantial margin compared with the basic seq2seq model.

2 Related Work

There has not been much work concerning computational TCM. [13] attempt to build a TCM clinical data warehouse so that the TCM knowledge can be analyzed and used. This is a typical way of collecting data, since the number of prescriptions given by the practitioners in the clinics is very large. However, in reality, most of the TCM doctors do not refer to the constructed digital systems, because the quality of the input data tends to be poor. Therefore, we choose prescriptions in the classics (books or documentation) of TCM. Although the available data can be fewer than the clinical data, it guarantees the quality of the prescriptions.

[11] attempt to construct a self-learning expert system with several simple classifiers to facilitate the TCM diagnosis procedure, [10] propose to use shallow neural networks and CRF based multi-labeling learning methods to model TCM inquiry process, but they only considered the disease of chronic gastritis and its

taxonomy is very simple. These methods either utilize traditional data mining methods or are highly involved with expert crafted systems.

3 Data Construction

When constructing our TCM herbal therapy dataset, we first considered the TCM medical records (中医医案) in the history, which contain a lot of good reference medical cases. The medical records are widely referenced by the doctors in the treatment. However, they have not been well digitalized, which makes it hard to extract the prescriptions out of the descriptive natural language text from the records. Another way to get large scale prescriptions is to collect from TCM clinics. The problem is that this kind of valuable data is not publicly available and the quality of such data is not good enough. Therefore, we turned to classic resources on the Internet finally.

We crawl the data from TCM Prescription Knowledge Base (中医方剂知识库) [4]. This knowledge base includes comprehensive TCM documentation in the history. The database includes 710 TCM historic books or documents as well as some modern ones, consisting of 85,166 prescriptions in total. Each item in the database provides the name, the origin, the composition, the effect, the contraindications, and the preparation method. We clean and formalize the database and get 82,044 usable symptom-prescription pairs

In the process of formalization, we temporarily omit the dose information and the preparation method description, as we are mainly concerned with the composition. Because the names of the herbs have evolved a lot, we conclude heuristic rules as well as specific projection rules to project some rarely seen herbs to their similar forms that are normally referred to. There are also prescriptions that refer to the name of other prescriptions. We simply substitute these names with their constituents.

To make the experiment result more robust, we conduct our experiments on two separate test datasets. The first one is a subset of the data described in Section 3. We randomly split the whole data into three parts, the training data (90%), the development data (5%) and the test data (5%). The second one is a set of symptom-prescription pairs we manually extracted from the modern **text book** of the course **Formulaology of TCM** (中医方剂学) that is popularly adopted by many TCM colleges in China.

There are more cases in the first sampled test dataset (4,102 examples), but it suffers from lower quality, as this dataset was parsed with simple rules, which may not cover all exceptions. The second test dataset has been proofread and all of the prescriptions are the most classical and influential ones in the history. So the quality is much better than the first one. However, the number of the cases is limited. There are 141 symptom-prescription pairs in the second dataset. Therefore, we use two test sets to do evaluation to take the advantages of both data magnitude and quality.

[4] `http://www.hhjfsl.com/fang/`

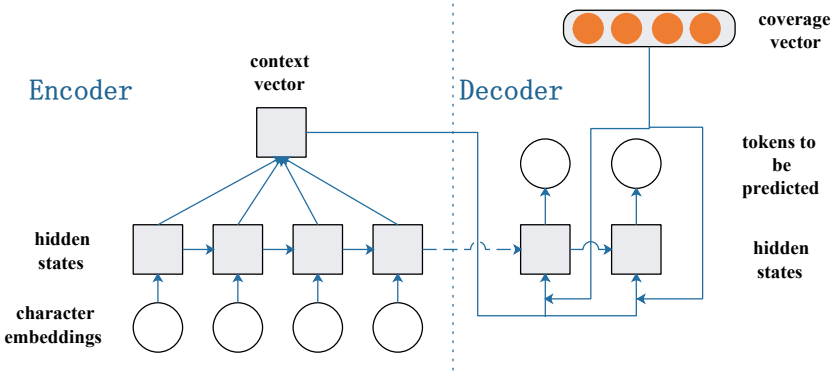


Fig. 1. An illustration of our model. The model is built on the basis of seq2seq model with attention mechanism. We use a coverage mechanism to reduce repetition problem.

4 Methodology

In this section, we first describe the definition of the TCM prescription generation task. Then, we propose how to enhance the seq2seq model with coverage mechanism and soft loss in the prescription composition task. A brief illustration of the our final model is shown in Figure 1.

4.1 Task Definition

Given a TCM herbal treatment dataset that consists of N data samples, the i-th data sample $(x^{(i)}, p^{(i)})$ contains one piece of source text $x^{(i)}$ that describes the symptoms, and M_i TCM herbs $(p_1^i, p_2^i, ..., p_{M_i}^i)$ that make up the herb prescription $p^{(i)}$.

We view the symptoms as a sequence of characters $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, ..., x_T^{(i)})$. We do not segment the characters into words because they are mostly in traditional Chinese that uses characters as basic semantic units. The herbs $p_1^i, p_2^i, ..., p_{M_i}^i$ are all different from each other.

4.2 Proposed Model

The model consists of two parts, an encoder and a decoder. The encoder is bound to take in the source sequence and compress the sequence into a series of hidden states. The decoder is used to generate a sequence of target tokens based on the information embodied in the hidden states given by the encoder.

In our TCM prescription generation task, the encoder RNN encodes the variable-length symptoms in character sequence $x = (x_1, x_2, ..., x_T)$ into a set of hidden representations $h = (h_1, h_2, ..., h_T)$, by iterating the following equations along time t:

$$h_t = GRU(x_t, h_{t-1}) \tag{1}$$

We choose the bidirectional version of recurrent neural networks as the encoder to solve the problem that the later words get more emphasis in the unidirectional version. We concatenate both the h_t in the forward and backward pass and get $\widehat{h_t}$ as the final representation of the hidden state at time step t.

The decoder is another RNN. It generates a variable-length sequence $y = (y_1, y_2, ..., y_{T'})$ token by token (herb), through a conditional language model:

$$s_t = GRU(s_{t-1}, c_t, Ey_{t-1}) \tag{2}$$

$$p(y_t|y_{1,...,t}, x) = g(s_t) \tag{3}$$

where s_t is the hidden state of the decoder RNN at time step t. The non-linear function g is a $softmax$ layer, which outputs the probabilities of all the herbs in the herb vocabulary. $E \in (V \times d)$ is the embedding matrix of the target tokens, V is the number of herb vocabulary, d is the embedding dimension. y_{t-1} is the last predicted token.

In the decoder, the context vector c_t is calculated based on the hidden state s_{t-1} of the decoder at time step $t-1$ and all the hidden states in the encoder. The procedure is known as the attention mechanism. The attention mechanism is expected to supplement the information from the source sequence that is more connected to the current hidden state of the decoder instead of only depending on a fixed vector produced by the encoder.

$$c_t = \sum_{j=1}^{T} \alpha_{tj} h_j \tag{4}$$

$$\alpha_{tj} = \frac{\exp\left(a\left(s_{t-1}, h_j\right)\right)}{\sum_{k=1}^{T} \exp\left(a\left(s_{t-1}, h_k\right)\right)} \tag{5}$$

The context vector c_t is calculated as a weighted sum of hidden representation produced by the encoder $\mathbf{h} = (h_1, ..., h_T)$. $a(s_{t-1}, h_j)$ is a soft alignment function that measures the relevance between s_{t-1} and h_j. It computes how much h_j is needed for the t-th output word based on the previous hidden state of the decoder s_{t-1}.

Different from natural language generation tasks, there is no duplicate herb in the TCM prescription generation task. When directly applying seq2seq model in this task, the decoder tends to generate some frequently observed herbs over and over again (see Section 5.5 for an example). Although we can prune the repeated herbs through post processing by eliminating the repeated ones, it still hurts the recall performance as the maximum length of a prescription is limited.

To encourage the decoder to generate more diverse and reasonable herb tokens, we propose to apply coverage mechanism to make the model aware of the already generated herbs. Coverage mechanism was first proposed by [8] to help the decoder focus on the part that has not been paid much attention by feeding a fertility vector to the attention calculation, indicating how much information of the input is used.

In our model, we do not use the fertility vector to tune the attention weights. The reason is that the symptoms are related to others and altogether describe the

whole disease, which is explained in Section 1. Still, inspired by its motivation, we adapt the coverage mechanism to the decoder where a coverage vector is fed to the GRU cell together with the context vector. Equation 2 is then replaced by the following ones.

$$a_t = \tanh(WD_t + b) \tag{6}$$

$$s_t = f(s_{t-1}, c_t, Ey_{t-1}, a_t) \tag{7}$$

where a_t is the coverage vector at the t-th time step in decoding. D_t is the one-hot representation of the generated tokens until the t-th time step. $W \in \mathbb{R}^{V \times H}$ is a learnable parameter matrix, where V is the size of the herb vocabulary and H is the size of the hidden state. By feeding the coverage vector, which is also a sketch of the generated herbs, to the GRU as part of the input, our model can softly switch more probability to the herbs that have not been predicted. This way, the model is encouraged to produce novel herbs rather than repeatedly predicting the frequently observed ones, thus increasing the recall rate.

We argue that even though the order of the herbs matters when generating the prescription [9], we should not strictly restrict the order. However, the traditional cross entropy loss function applied to the basic seq2seq model puts a strict assumption on the order of the labels. To deal with the task of predicting weakly ordered labels (or even unordered labels), we propose a soft loss function instead of the original hard cross entropy loss function:

$$loss = -\sum_t q'_t \; log(p_t) \tag{8}$$

Instead of using the original hard one-hot target probability q_t, we use a soft target probability distribution q'_t, which is calculated according to q_t and the target sequence $\boldsymbol{q}$ of this sample. Let $\boldsymbol{q_v}$ denote the bag of words representation of $\boldsymbol{q}$, where only slots of the target herbs in $\boldsymbol{q}$ are filled with $1s$. We use a function ξ to project the original target label probability q_t into a new probability distribution q'_t.

$$q'_t = \xi(q_t, \boldsymbol{q_v}) \tag{9}$$

The function ξ is designed so as to decrease the harsh punishment when the model predicts the labels in the wrong order. In this paper, we apply a simple yet effective projection function as Equation 10. This is an example implementation, and one can design more sophisticated projection functions if needed.

$$\xi(y_t, \boldsymbol{s}) = ((\boldsymbol{q_v}/M) + y_t)/2 \tag{10}$$

where M is the length of q. This function means that at the t-th time of decoding, for each target herb token p_i, we first split a probability density of 1.0 equally across all the l herbs into $1/M$. Then, we take the average of this probability distribution and the original probability q_t to be the final probability distribution at time t.

Data	Average Length	Max Length	Length Under 20
Crawled Data	7.2	108	97.99%
Textbook Data	6.7	16	100%

Table 2. The statistic of the length of prescriptions. Crawled data means the overall data crawled from the Internet. Length Under 20 means the percentage of data that are shorter or equal than 20.

5 Experiment

5.1 Proposed Multi-label Baseline

In this part, we present the **Multi-label** baseline we apply. In this model, we use a BiGRNN as the encoder, which encodes symptoms in the same way as it is described in Section 4. Because the position of the herbs does not matter in the results, for the generation part, we implement a multi-label classification method to predict the herbs. We add a softmax layer above the vector of the symptoms to predict the probability of each herb. We use the multi-label max-margin loss (MultiLabelMarginLoss in pytorch) as the optimization objective, because this loss function is more insensitive to the threshold. We set the threshold to be 0.5, that is, if the probability given by the model is above 0.5 and within the top 20 (same to seq2seq model). The way to calculate probability is shown below.

During evaluation, we choose the herbs satisfying two conditions: 1) The predicted probability of the herb is within top 20 (same to seq2seq) among all the herbs; 2) The predicted probability is above a threshold 0.5.

5.2 Experiment Settings

We set the embedding size of both Chinese characters in the symptoms and the herb tokens to 100. We set the hidden state size to 300, and the batch size to 20. We set the maximum length of the herb sequence to 20 because the length of nearly all the prescriptions are within this range (see Table 2 for the statistics of the length of prescriptions). Adam [3] is adopted to optimize the parameters. We tune all the hyper-parameters on the development set, and use the model parameters that generate the best F1 score on the development set in testing

5.3 Human Evaluation

Since medical treatment is a very complex task, we invite two professors from Beijing University of Chinese Medicine, which is one of the best Traditional Chinese Medicine academies in China. Both of the professors enjoy over five years of practicing traditional Chinese medical treatment. The evaluators are asked to evaluate the prescriptions with scores between 0 and 10. Both the textual symptoms and the standard reference are given, which is similar to the form of evaluation in a normal TCM examination. The Pearson's correlation coefficient between the two evaluators is 0.72 and the Spearman's correlation

Model	Evaluator 1	Evaluator 2	Average Score
Multi-Label	4.5	4.1	4.3
Basic seq2seq	6.8	6.6	6.7
Proposal	7.4	7.1	7.3

Table 3. Professional evaluation on the Textbook test set. The score is in range 0∼10.

Model	Crawled set			Textbook set		
	Precision	Recall	F1	Precision	Recall	F1
Multi-label	10.83	29.72	15.87	13.51	40.49	20.26
Basic seq2seq	26.03	13.52	17.80	30.97	23.70	26.85
Proposal	29.57	17.30	21.83	38.22	30.18	33.73

Table 4. Automatic evaluation results of different models on the two test datasets.

coefficient is 0.72. Both p-values are less than 0.01, indicating strong agreement. Different from the automatic evaluation method, the human evaluators focus on the potential curative effect of the candidate answers, rather than merely the literal similarity, which is more reasonable and close to reality.

Because the evaluation procedure is very time consuming (each item requires more than 1 minute), we only ask the evaluators to judge the results from Textbook test set. As is shown in Table 3, both of the basic seq2seq model and our proposed modification are much better than the multi-label baseline. Our proposed model gets a high score of 7.3, which can be of real help to TCM practitioners when prescribing in the real life treatment.

5.4 Automatic Evaluation Results

We use micro Precision, Recall, and F1 score as the automatic metrics to evaluate the results, because the internal order between the herbs does not matter when we do not consider the prescribing process. In Table 4, we show the results. One thing that should be noted is that since the data in Textbook test set have much better quality than crawled test set, the performance on Textbook test set is much higher than it is on Crawled test set, which is consistent with our instinct.

From the experiment results we can see that the baseline model multi-label has higher micro recall rate 29.72, 40.49 but much lower micro precision 10.83, 13.51. This is because unlike the seq2seq model that dynamically determines

Model	Crawled set			Textbook set		
	Precision	Recall	F1	Precision	Recall	F1
Basic seq2seq	26.03	13.52	17.80	30.97	23.70	26.85
+ coverage	26.69	12.88	17.37	37.09	24.12	29.23
+ soft loss	29.30	17.26	21.72	37.90	27.63	31.96
+ coverage & soft loss	29.57	17.30	21.83	38.22	30.18	33.73

Table 5. Ablation results of applying coverage mechanism and soft loss function.

Model	Original Total Length	Pruned Total Length	# Repetition
Basic seq2seq	859	716	143
+ coverage	724	609	115
+ soft loss	741	685	56
Proposal	782	743	39

Table 6. Prescription length of different models on Textbook test set.

the length of the generated sequence, the output length is rigid and can only be determined by thresholds. We take the tokens within the top 20 as the answer for the multi-label model.

As to the basic seq2seq model, although it beats the multi-label model overall, the recall rate drops substantially. This problem is partly caused by the repetition problem, the basic seq2seq model sometimes predicts high frequent tokens instead of more meaningful ones. Apart from this, although the seq2seq based model is better able to model the correlation between target labels, it makes a strong assumption on the order of the target sequence. In the prescription generation task, the order between herb tokens are helpful for generating the sequence. However, since the order between the herbs does not affect the effect of the prescription, we do not consider the order when evaluating the generated sequence. The much too strong assumption on order can hurt the performance of the model when the correct tokens are placed in the wrong order.

In Table 5 we show the effect of applying coverage mechanism and soft loss function. Coverage mechanism gives a sketch on the generated prescription. The mechanism not only encourages the model to generate novel herbs but also enables the model to generate tokens based on the already predicted ones. This is proved by the improvement on Textbook test set, where both the precision and the recall are improved over the basic seq2seq model. The most significant improvement comes from applying the soft loss function. The soft loss function can relieve the strong assumption of order made by seq2seq model. Because predicting a correct token in the wrong position is not as harmful as predicting a completely wrong token. This simple modification gives a big improvement on both test sets for all the three evaluation metrics.

In Table 6 we show the total length of generated prescriptions. We choose the Textbook test set because it enjoys better quality. From the table, we can also observe that applying coverage mechanism and soft loss function can greatly reduce the repetition number compared with the basic seq2seq model.

5.5 Case Study

In this part, we show an example generated by various models in Table 7. We choose the example in Textbook test set because the quality of Textbook test set is much more satisfactory. The multi-label model produces too many herbs that lower the precision.

For the basic seq2seq model, the result is better than multi-label baseline. In this case, most of the herbs can be matched with certain symptoms in the textual

Table 7. Actual predictions made by various models in Textbook test set. Multi-label model generates too many herb tokens, so we do not list all of them here. Reference is the standard answer prescription given by the text book.

Symptoms	外感风寒表虚证。恶风发热，汗出头疼，鼻鸣咽干，苔白不渴，脉浮缓或浮弱。
Translation	Exogenous wind-cold exterior deficiency syndrome. Aversion to wind, fever, sweating, headache, nasal obstruction, dry throat, white tongue coating, not thirsty, floating slow pulse or floating weak pulse.
Reference	桂枝 芍药 甘草 生姜 大枣
Multi-label	防风 知母 当归 川芎 黄芪 橘红 甘草 茯苓 白术 葛根 荆芥 柴胡 麦冬 泽泻 车前子 石斛 木通 赤茯苓 升麻 白芍药
Basic seq2seq	柴胡 干葛 川芎 桔梗 甘草 陈皮 半夏
Proposal	桂枝 麻黄 甘草 生姜 大枣

description.[5] However, the problem is that unlike the reference, the composition of herbs lacks the overall design. The symptoms should not be treated independently, as they are connected to other symptoms. For example, the appearance of symptom "头疼" (headache) must be treated together with "汗出" (sweat). When there is simply headache without sweat, "川芎" (Ligusticum wallichii) may be suitable. However, since there is already sweat, this herb is not suitable in this situation. This drawback results from the fact that this model heavily relies on the attention mechanism that tries to match the current hidden state in the decoder to a part of the context in the encoder.

For our proposed model, the results are much more satisfactory. "外感风寒" (Exogenous wind-cold exterior deficiency syndrome) is the reason of the disease, the symptoms "恶风发热，汗出头疼，鼻鸣咽干，苔白不渴，脉浮缓或浮弱" (Aversion to wind, fever, sweating, headache, nasal obstruction, dry throat, white tongue coating, not thirsty, floating slow pulse or floating weak pulse) are the corresponding results. The prescription generated by our proposed model can also be used to cure "外感风寒" (Exogenous wind-cold exterior deficiency syndrome), in fact "麻黄" (Chinese ephedra) and "桂枝" (cassia twig) together is a common combination to cure cold. However, "麻黄" (Chinese ephedra) is not suitable here because there is already sweat. One of the most common effect of "麻黄" (Chinese ephedra) is to make the patient sweat. Since there is already sweat, it should not be used. Compared with the basic seq2seq model, our proposed model have a sense of overall disease, rather than merely discretely focusing on individual symptoms.

[5] "柴胡" (radix bupleuri)、"葛根" (the root of kudzu vine) can be roughly matched with "恶风发热，汗出头疼" (Aversion to wind, fever, sweating, headache), "甘草" (Glycyrrhiza)、"陈皮" (dried tangerine or orange peel)、"桔梗" (Platycodon grandiflorum) can be roughly matched with "鼻鸣咽干，苔白不渴" (nasal obstruction, dry throat, white tongue coating, not thirsty), "川芎" (Ligusticum wallichii) can be used to treat the symptom of "头疼" (headache).

6 Conclusion

In this paper, we propose a TCM prescription generation task that automatically predicts the herbs in a prescription based on the textual symptom descriptions. To our knowledge, this is the first time that this task is considered. To advance the research in this task, we construct a large dataset. Besides the automatic evaluation, we also invite professionals to evaluate the prescriptions given by various models, the results of which show that our model reaches the score of 7.3 out of 10, demonstrating the effectiveness.We hope this work can lay a foundation and encourage more researchers to pay attention to the automatic TCM prescription generation problem.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1724–1734. Association for Computational Linguistics (2014). https://doi.org/10.3115/v1/D14-1179, http://www.aclweb.org/anthology/D14-1179
3. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
4. Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., Jurafsky, D.: Deep reinforcement learning for dialogue generation. arXiv preprint arXiv:1606.01541 (2016)
5. Li, J., Monroe, W., Shi, T., Ritter, A., Jurafsky, D.: Adversarial learning for neural dialogue generation. arXiv preprint arXiv:1701.06547 (2017)
6. See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368 (2017)
7. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)
8. Tu, Z., Lu, Z., Liu, Y., Liu, X., Li, H.: Modeling coverage for neural machine translation. arXiv preprint arXiv:1601.04811 (2016)
9. Vinyals, O., Bengio, S., Kudlur, M.: Order matters: Sequence to sequence for sets. arXiv preprint arXiv:1511.06391 (2015)
10. Wang, L.: TCM inquiry modelling research based on Deep Learning and Conditional Random Field multi-lable learning methods. Ph.D. thesis, East China University of Science and Technology (2013)
11. Wang, X., Qu, H., Liu, P., Cheng, Y.: A self-learning expert system for diagnosis in traditional chinese medicine. Expert systems with applications **26**(4), 557–566 (2004)
12. Yin, J., Jiang, X., Lu, Z., Shang, L., Li, H., Li, X.: Neural generative question answering. arXiv preprint arXiv:1512.01337 (2015)
13. Zhou, X., Chen, S., Liu, B., Zhang, R., Wang, Y., Li, P., Guo, Y., Zhang, H., Gao, Z., Yan, X.: Development of traditional chinese medicine clinical data warehouse for medical knowledge discovery and decision support. Artificial Intelligence in medicine **48**(2), 139–152 (2010)