Solving Chinese Character Puzzles Based on Character Strokes

Da Ren[1], Yi Cai[1(✉)], Weizhao Li[1], Ruihang Xia[1], Zilu Li[2], and Qing Li[3]

[1] South China University of Technology, Guangdong, China
ycai@scut.edu.cn
[2] Guangzhou Tianhe Foreign Language School, Guangdong, China
[3] The Hong Kong Polytechnic University, Hong Kong, China

Abstract. Chinese character puzzles are popular games in China. To solve a character puzzle, people need to fully consider the meaning and the strokes of each character in puzzles. Therefore, Chinese character puzzles are complicated and it can be a challenging task in natural language processing. In this paper, we collect a Chinese character puzzles dataset (CCPD) and design a Stroke Sensitive Character Guessing (SSCG) Model. SSCG can consider the meaning and strokes of each character. In this way, SSCG can solve Chinese character puzzles more accurately. To the best of our knowledge, it is the first work which tries to handle the Chinese character puzzles. We evaluate SSCG on CCPD. The experiment results show the effectiveness of the SSCG.

Keywords: Chinese · Character puzzles · Character strokes

1 Introduction

Chinese character puzzles have a long history in China. A Chinese character puzzle is always short and the answer is a single Chinese character. To solve a Chinese character puzzle, people have to fully understand the meaning of each character. Moreover, they also need to make use of strokes of characters in puzzles. Therefore, it is not easy to solve a Chinese character puzzle.

We show examples of Chinese character puzzles in Fig. 1. In Fig. 1 (a), there are two characters "日" and "月". To solve this puzzle, people need to combine these two characters to be a new character " 明". Character strokes are important in solving this puzzle, since the strokes of the answer all come from the strokes in the puzzle.

In Fig. 1 (b), people have to fully understand the meaning of each character to solve this puzzle. The word " 儿子(son)" indicates "男(man)" and the meaning of the word “出世(give birth to)” is same with the meaning of “ 生(deliver)”. Therefore, the answer is the combination of “生” and “男” which is “甥”.

Fig. 1 (c) is a more complicated puzzle. There is a character “泼” in the puzzle. The phrase “ 泼水节” indicates “Water-Splashing Festival”. In this festival, people splash water. Therefore, We need to remove “ 水(water)” from the character “泼”. “氵 ” also indicates water in Chinese. After removing “氵 ” from “ 泼”, the answer is “ 发”. In this puzzle, people need to consider both the strokes and the meaning of the characters.

(a) 日月各西东 ⟶ 日 + 月 ⟶ 明

(b) 儿子出世 ⟶ 生 + 男 ⟶ 甥

(c) 泼水节 ⟶ 泼 − 水 ⟶ 发

Fig. 1. Examples of Chinese Character Puzzles

Overall, Chinese character puzzles are complicated. To solve the puzzles, people need to fully consider the meaning and the strokes of each character. We consider that it can be a challenging task in natural language processing. Therefore, we collect a number of Chinese character puzzles and their answers. What’s more, we propose a Stroke Sensitive Character Guessing (SSCG) Model. SSCG can solve Chinese character puzzles by considering both character meanings and strokes.

Our contributions can be summarized as follows:

– We propose a Stroke Sensitive Character Guessing (SSCG) Model. To the best of our knowledge, it is the first model which is designed to solve Chinese character puzzles. SSCG can solve the puzzles based on both characters and their strokes.
– We collect Chinese character puzzles to construct a Chinese character puzzles dataset (CCPD). CCPD can support further research of Chinese character puzzles.
– We conduct experiments to evaluate the performance of SSCG on CCPD. The experiment results show the effectiveness of the SSCG.

2 Related Work

To the best of our knowledge, no research has been addressed on Chinese character puzzles. In this work, we regard character puzzles as retrieval tasks. Therefore, we introduce researches which are based on retrieval models.

2.1 Retrieval-based Question Answering

Early researches on answer selection generally treat this task as statistic classification problems. These methods [9, 14, 24] rely on exploring various feature as the representation of question answering. However, these methods rely heavily on feature engineering, which requires a large amount of manual work and domain expertise.

Recently, researchers propose a number of data-driven models. Wu et al. [17] introduce a gate mechanism to model the interactions between question and answer. Their model can aggregate more relevant information to identify the relationship between questions and answers. Wu et al. [16] further utilize the subject-body relationship of question to condense question representation, where the multi-dimensional attention mechanism is adopted.

2.2 Retrieval-based Conversation Systems

Conversation systems can be traced back to Turing Test [11]. Models in conversation systems can generally be divided into two categories: generation-based methods [8, 10, 19, 23] and retrieval-based methods [3, 4]. Generation-based methods generate a response according to a conversation context. Retrieval-based methods retrieve a response from a pre-defined repository [6, 20].

Early studies of retrieval-based methods focus on response selection for single-turn conversation [4, 13, 15]. Recently researchers begin to focus on multi-turn conversation [7, 21, 22, 27]. A number of methods are proposed to improve the performance of retrieval models [18, 25, 26, 28]. Wu et al. [21] propose a sequential matching network to capture the important contextual information. Young et al. [25] investigate the impact of providing commonsense knowledge about the concepts covered in the dialogue. Inspired by Transformer [12], Zhou et al. [28] investigate matching a response with its multi-turn context using dependency information based entirely on attention.

3 Model

To solve Chinese character puzzles, we propose a Stroke Sensitive Character Guessing (SSCG) Model (as shown in Fig. 2). There are three modules in SSCG: Answer Stroke Encoder (ASE), Puzzle Stroke Encoder (PSE) and Puzzle Solver (PS). ASE and PSE encode the strokes in answers and puzzles as vectors respectively. Then, PS gives a matching score between puzzles and their candidate answers.

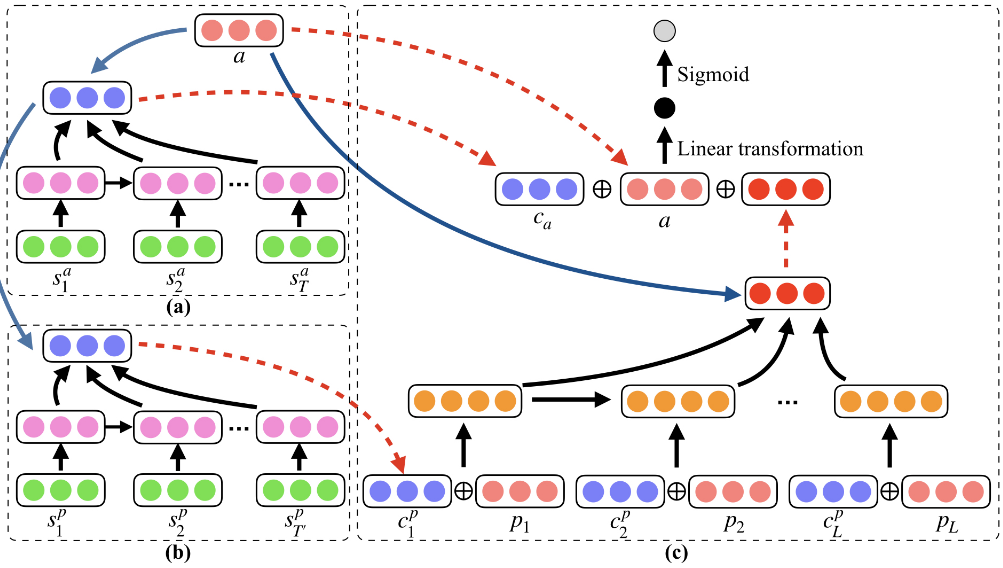

Fig. 2. The General Structure of SSCG

3.1 Problem Definition

Given a Chinese character puzzle $P = (p_1, p_2, ..., p_L)$ and a set of candidate answers $A = \{a_1, a_2, ..., a_N\}$, our task is to find the correct answer of P from A. Both p_l and a_l are Chinese character. L is the length of the puzzle and N is the size of the candidate set.

Answer Stroke Encoder Strokes in answers are always important in solving Chinese character puzzles. Therefore, we propose an Answer Stroke Encoder (ASE) to encode the information from answer strokes into fixed length vectors (as shown in Fig. 2 (a)).

Given an answer a and its strokes $S_a = (s_1^a, s_2^a, ..., s_T^a)$ (T is the size of S_a), we use Recurrent Neural Networks (RNNs) to construct the answer stroke encoder. It is described in Equation 1.

$$h_t^a = f_{GRU}(h_{t-1}^a, e(s_t^a)) \tag{1}$$

where h_t^a is the hidden state of the t-th timestep, s_t^a is the t-th stroke, $e(\cdot)$ is the embedding of the stroke, $f_{GRU}(\cdot)$ means Gated Recurrent Unit (GRU) [2]. Then we use an attention mechanism [1] to calculate the weighted sum of the hidden states.

$$
\begin{aligned}
c_a &= \sum_{i=1}^{T} \alpha_i h_i^a \\
\alpha_i &= \frac{exp(\beta_i)}{\sum_{j=1}^{T} exp(\beta_j)} \\
\beta_i &= W_a tanh(W_b(e'(a) \oplus h_i^a))
\end{aligned} \tag{2}
$$

where $e'(a)$ indicates the character embedding of answer a, h_i^a is the i-th hidden state, $\oplus$ is a concatenation operation. W_a and W_b are weighted matrices to be learned.

Puzzle Stroke Encoder According to our observations, the strokes of each character are important in solving Chinese character puzzles. Moreover, the strokes in the characters in a puzzle are always related the strokes in its answer. Thus, in Puzzle Stroke Encoder (PSE), we encode the strokes of each character with the guidance of the information from answer strokes (as shown in Fig. 2 (b)). Giving a character in a puzzle character p and its strokes $S_p = (s_1^p, s_2^p, ..., s_{T'}^p)$, we use RNNs to encode the strokes which is described in Equation 3.

$$
h_t^p = f_{GRU}(h_{t-1}^p, e(s_t^p)) \tag{3}
$$

where h_t^p is the hidden state in the t-th timestep, s_t^p is the t-th stroke of p, $e(\cdot)$ is the embedding of the corresponding stroke.

We use an attention mechanism to combine the hidden states. The attention mechanism we used is described in Equation 4.

$$
\begin{aligned}
c_p &= \sum_{i=1}^{T'} \alpha'_i h_i^p \\
\alpha'_i &= \frac{exp(\beta'_i)}{\sum_{j=1}^{T'} exp(\beta'_j)} \\
\beta'_i &= W_c tanh(W_d(c_a \oplus h_i^p))
\end{aligned} \tag{4}
$$

where c_a is calculated by Equation 2, h_i^p is the i-th hidden states. W_c and W_d are weighted matrices to be learned. We use c_p as the stroke representation of character p.

Puzzle Solver In Puzzle Solver (PS), we use an RNN to encode the information of each character in a puzzle. We represent each character with the concatenation of its character embedding and stroke representation. This process is described in the following.

$$h_t^s = f_{GRU}(h_{t-1}^s, e'(p_t) \oplus c_p^t) \tag{5}$$

where h_t^s is the hidden state in the t-th timestep, p_t is the t-th word in the puzzle, $e'(\cdot)$ represents the embedding of a character, c_p^t is the stroke representation of p_t and it is calculated according to Equation 4.

Then we calculate the weighted sum of the hidden states according to Equation 6.

$$\begin{aligned} c_s &= \sum_{i=1}^{T'} \hat{\alpha}_i h_i^s \\ \hat{\alpha}_i &= \frac{exp(\hat{\beta}_i)}{\sum_{j=1}^{T'} exp(\hat{\beta}_j)} \\ \hat{\beta}_i &= W_e tanh(W_f(e'(a) \oplus h_i^s)) \end{aligned} \tag{6}$$

where $e'(a)$ is the embedding of a, h_i^s is the hidden state in the i-th timestep. W_e and W_f are weighted matrices to be learned.

As shown in Fig. 2 (c), we combine $e'(a)$, c_a and c_s. Then, we calculate the matching scores between the puzzle and the answer according to Equation 7.

$$\hat{s} = \sigma(W_g(e'(a) \oplus c_a \oplus c_s)) \tag{7}$$

where $\sigma(\cdot)$ is a sigmoid function, W_g is a weighted matrix to be learned.

In training process, we use binary cross entropy as our loss function. It is calculated according to Equation 8.

$$Loss = -\frac{1}{M} \sum_{i=1}^{M} (y_i log(\hat{s}_i) + (1 - y_i) log(1 - \hat{s}_i)) \tag{8}$$

where M is the batch size, $\hat{s}_i$ is the i-th matching score calculated by Equation 7, y_i is the target. y_i is 1 when the answer is correct and it is 0 when the answer is wrong.

In test process, SSCG gives score to all candidate answers and we rerank the candidate answers according to their matching scores.

4 Experiment

4.1 Dataset

We collect Chinese character puzzles from *Baidu Hanyu*[4] and *Hydcd*[5]. Each character puzzle has a corresponding answer. The strokes of each word is collected from

[4] https://hanyu.baidu.com

[5] http://www.hydcd.com/baike/zimi.htm

Httpcn[6]. All the Chinese character puzzles contain 2,738 different characters. The length of puzzles ranges from 1 to 38. We finally choose 9,354 puzzle-answer pairs as training set, 500 pairs as validation set and 450 pairs as test set. The statistics of the dataset is shown in Table 1. This dataset is available online[7].

Table 1. Data Statistics

	Train	Valid	Test
Avg.# characters per puzzle	6.44	5.75	5.86
Avg.# strokes per character in puzzle	9.30	8.27	8.14
Different characters in puzzle	2662	879	821

4.2 Experiment setup

Our model is implemented with *PyTorch*[8]. In practice, we initialize character embedding randomly. We do not share the character embedding between puzzles and answers. In training set, we choose 2,687 characters in puzzles as the puzzle vocabulary, 3,198 characters in answers as the answer vocabulary. Characters in puzzles but not in the puzzle vocabulary and the characters in answers but not in the answer vocabulary are replaced with <unk>.

We set the word embedding size as 128. The RNNs in ASE, PSE and PS are 1-layer RNNs and the hidden size is set to be 256. We share the parameters of the RNNs in ASE and PSE. We use the Adam [5] as our optimizer. The batch size is set to be 128. We set the learning rate as $1e-04$. The dropout rate is set to be 0.1.

4.3 Evaluation Metric

$R_k@E$ In this paper, we use $R_k@E$ to evaluate the performance of compared models automatically. For each puzzles in test set, there are k different characters in candidate answer set. We rank the candidate answers by the score given by models. If the correct answer is ranked in top E, this answer will be correct. In our experiments, we use $R_2@1$, $R_5@1$, $R_{10}@1$ to evaluate the performance of models.

[6] http://hy.httpcn.com
[7] https://github.com/wizare/A-Chinese-Character-Puzzles-Dataset
[8] https://pytorch.org

4.4 Compared Model

To the best of our knowledge, there is no existing model about Chinese character puzzles. We design compared models in the following.

Plain Guessing Model (PGM) We concatenate a character puzzle and a candidate answer character as an input sequence. We use RNNs with GRU to process the sequence. We use self-attention and max-pooling to combine the hidden states. Then we use a linear transformation and sigmoid function to calculate the matching score.

Character Guessing Model (CGM) In CGM, we remove the ASE and PSE in SSCG. CGM calculates the matching score only based on word embeddings. We compare the performance between CGM and SSCG to explore the effectiveness of character strokes.

4.5 Experiment Result

Table 2. Experiment Results

Model	$R_2@1$	$R_5@1$	$R_{10}@1$
PGM	52.68%	28.82%	15.54%
CGM	54.98%	36.14%	29.92%
SSCG	**57.44%**	**38.06%**	**32.36%**

The experiment results are shown in Table 2. According to our experiments, CGM significantly outperforms PGM (p-value<0.05). PGM gets 52.68% in $R_2@1$, while the value of CGM is 54.98%. In $R_5@1$, CGM gets 36.14% which is 7.32% higher than the value of PGM. The $R_{10}@1$ of PGM is 15.54% which is 14.38% lower than CGM. In PGM, there is only a self-attention and max-pooling operation. In CGM, all the hidden states are summed together under the guidance of the candidate answers. Therefore, CGM can find the answers more accurately since it can focus on important characters.

Moreover, SSCG is significantly outperforms PGM (p-value<0.05) and CGM (p-value<0.01) in all evaluation metrics. In $R_2@1$, SSCG gets 57.44% which is 2.46% higher than CGM. The $R_5@1$ of SSCG is 38.06% and it is 1.92% higher than CGM. SSCG gets 32.36% in $R_{10}@1$, while the value of CGM is 29.92%. There are ASE and PSE in SSCG. Both of them can help SSCG to be sensitive the character strokes. As a result, SSCG can solve character puzzles better.

ID	Puzzle		Explanation		Target	PGM	CGM	SSCG
(1)	归卧南山陲	⟶	山 + 归	⟶	岿	裹	岿	岿
(2)	河灯半明灭	⟶	水 + 丁	⟶	汀	羊	汀	汀
(3)	二大王心如刀绞	⟶	二王 + 心 + 绞	⟶	瑟	交	交	瑟
(4)	甘心独自归	⟶	甘 − 心	⟶	廿	仙	仙	廿

Fig. 3. Results of Compared Models

To further compare the performance between PGM, CGM and SSCG, we sample some cases and show in Fig. 3. In case 1, we need to combine the character “归” and “山” together. The result is “岿”. PGM fails to answer it correctly while both CGM and SSCG give a correct answer. In case 2, we need to extract a part of the character “河” and “ 灯” to get the result “汀”. Both CGM and SSCG can answer it correctly. However, PGM gives an incorrect answer “羊”. The puzzles in these two cases can be solved by combining two characters. CGM and SSCG can answer them correctly. It shows the effectiveness of the attention mechanisms in these two models. The puzzles in case 3 and 4 are more complicated. In case 3, a model needs to extract “二王”, “心” and combining the meaning of “绞” to solve this puzzle. Both PGM and CGM fail to solve this puzzle. However, SSCG can successfully get the correct result “ 瑟”. According to the puzzle in case 4, the “ 心(heart)” of “甘” should be removed. The word “心(heart)” indicates the “-” in the character “甘”. Thus, the result should be “ 廿”. In this case, only SSCG gives a correct answer. After considering the meanings and strokes of characters, SSCG can solve more complicated Chinese character puzzles.

5 Conclusion

In this paper, we propose a Stroke Sensitive Character Guessing (SSCG) Model which can solve Chinese character puzzles. We collect a Chinese character puzzle dataset. We conduct experiments to demonstrate the effectiveness of the attention mechanism and the strokes. Experiment results show that the attention mechanism and the stroke encoders (ASE and PSE) can significantly improve the performance.

In the future, we will try to further improve the model so that it can get a better performance. We plan to take the advantage of knowledge graph into our model for even better performance.

Acknowledgment

This work presented in this paper is partially supported by the Fundamental Research Funds for the Central Universities, SCUT (Nos. 2017ZD048, D2182480), the Tiptop Scientific and Technical Innovative Youth Talents of Guangdong special support program (No.2015TQ01X633), the Science and Technology Planning Project of Guangdong Province (No.2017B050506004), the Science and Technology Program of Guangzhou (Nos. 201704030076, 201802010027). The research described in this paper has been supported by a collaborative research grant from the Hong Kong Research Grants Council (project no. C1031-18G).

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings [1]
2. Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. In: Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014. pp. 103–111 (2014)
3. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. pp. 2042–2050 (2014)
4. Ji, Z., Lu, Z., Li, H.: An information retrieval approach to short text conversation. CoRR **abs/1408.6988** (2014)
5. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
6. Lowe, R., Pow, N., Serban, I., Pineau, J.: The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In: Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic. pp. 285–294 (2015)
7. Lowe, R., Pow, N., Serban, I., Pineau, J.: The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In: Proceedings of the SIGDIAL 2015 Conference, The 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 2-4 September 2015, Prague, Czech Republic. pp. 285–294 (2015)

8. Ren, D., Cai, Y., Lei, X., Xu, J., Li, Q., Leung, H.: A multi-encoder neural conversation model. Neurocomputing **358**, 344–354 (2019). https://doi.org/10.1016/j.neucom.2019.05.071
9. Severyn, A., Moschitti, A.: Automatic feature engineering for answer selection and extraction. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL. pp. 458–467 (2013)
10. Shao, Y., Gouws, S., Britz, D., Goldie, A., Strope, B., Kurzweil, R.: Generating high-quality and informative conversation responses with sequence-to-sequence models. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017. pp. 2210–2219 (2017)
11. Turing, A.M.: Computing machinery and intelligence. Mind **59**(236), 433–460 (1950)
12. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA. pp. 6000–6010 (2017)
13. Wang, H., Lu, Z., Li, H., Chen, E.: A dataset for research on short-text conversations. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL. pp. 935–945 (2013)
14. Wang, M., Smith, N.A., Mitamura, T.: What is the jeopardy model? A quasi-synchronous grammar for QA. In: EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic. pp. 22–32 (2007)
15. Wang, M., Lu, Z., Li, H., Liu, Q.: Syntax-based deep matching of short texts. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015. pp. 1354–1361 (2015)
16. Wu, W., Sun, X., Wang, H.: Question condensing networks for answer selection in community question answering. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers. pp. 1746–1755 (2018)
17. Wu, W., Wang, H., Li, S.: Bi-directional gated memory networks for answer selection. In: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data - 16th China National Conference, CCL 2017, - and - 5th International Symposium, NLP-NABD 2017, Nanjing, China, October 13-15, 2017, Proceedings. pp. 251–262 (2017). https://doi.org/10.1007/978-3-319-69005-6_21
18. Wu, Y., Li, Z., Wu, W., Zhou, M.: Response selection with topic clues for retrieval-based chatbots. Neurocomputing **316**, 251–261 (2018). https://doi.org/10.1016/j.neucom.2018.07.073

19. Wu, Y., Wei, F., Huang, S., Wang, Y., Li, Z., Zhou, M.: Response generation by context-aware prototype editing. In: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. pp. 7281–7288 (2019)
20. Wu, Y., Wu, W., Xing, C., Zhou, M., Li, Z.: Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers. pp. 496–505 (2017). https://doi.org/10.18653/v1/P17-1046
21. Wu, Y., Wu, W., Xing, C., Zhou, M., Li, Z.: Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers. pp. 496–505 (2017). https://doi.org/10.18653/v1/P17-1046
22. Yan, R., Song, Y., Wu, H.: Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, SIGIR 2016, Pisa, Italy, July 17-21, 2016. pp. 55–64 (2016). https://doi.org/10.1145/2911451.2911542
23. Yao, K., Peng, B., Zweig, G., Wong, K.: An attentional neural conversation model with improved specificity. CoRR **abs/1606.01292** (2016)
24. Yih, W., Chang, M., Meek, C., Pastusiak, A.: Question answering using enhanced lexical semantic models. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers. pp. 1744–1753 (2013)
25. Young, T., Cambria, E., Chaturvedi, I., Zhou, H., Biswas, S., Huang, M.: Augmenting end-to-end dialogue systems with commonsense knowledge (2018)
26. Zhang, Z., Li, J., Zhu, P., Zhao, H., Liu, G.: Modeling multi-turn conversation with deep utterance aggregation. In: Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018. pp. 3740–3752 (2018)
27. Zhou, X., Dong, D., Wu, H., Zhao, S., Yu, D., Tian, H., Liu, X., Yan, R.: Multi-view response selection for human-computer conversation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016. pp. 372–381 (2016)
28. Zhou, X., Li, L., Dong, D., Liu, Y., Chen, Y., Zhao, W.X., Yu, D., Wu, H.: Multi-turn response selection for chatbots with deep attention matching network. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers. pp. 1118–1127 (2018)