

Solving and Generating Chinese Character Riddles

Chuanqi Tan[†] * Furu Wei[‡] Li Dong⁺ Weifeng Lv[†] Ming Zhou[‡]

[†]State Key Laboratory of Software Development Environment, Beihang University, China

[‡]Microsoft Research Asia

⁺University of Edinburgh

[†]tanchuanqi@nlsde.buaa.edu.cn ⁺li.dong@ed.ac.uk

[‡]{fuwei, mingzhou}@microsoft.com [†]lwf@buaa.edu.cn

Abstract

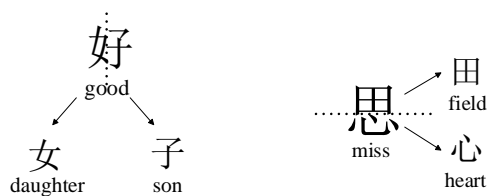
Chinese character riddle is a riddle game in which the riddle solution is a single Chinese character. It is closely connected with the shape, pronunciation or meaning of Chinese characters. The riddle description (sentence) is usually composed of phrases with rich linguistic phenomena (such as pun, simile, and metaphor), which are associated to different parts (namely radicals) of the solution character. In this paper, we propose a statistical framework to solve and generate Chinese character riddles. Specifically, we learn the alignments and rules to identify the metaphors between phrases in riddles and radicals in characters. Then, in the solving phase, we utilize a dynamic programming method to combine the identified metaphors to obtain candidate solutions. In the riddle generation phase, we use a template-based method and a replacement-based method to obtain candidate riddle descriptions. We then use Ranking SVM to rerank the candidates both in the solving and generation process. Experimental results in the solving task show that the proposed method outperforms baseline methods. We also get very promising results in the generation task according to human judges.

1 Introduction

The riddle is regarded as one of the most unique and vital elements in traditional Chinese culture, which is usually composed of a riddle description

*The work was done when the first author and the third author were interns at Microsoft Research Asia.

and a corresponding solution. The character riddle is one of the most popular forms of various riddles in which the riddle solution is a single Chinese character. While English words are strings of letters together, Chinese characters are composed of radicals that associate with meaning or metaphor. In other words, Chinese characters are usually positioned into some common structures, such as upper-lower structure, left-right structure, inside-outside structure, which means they can be decomposed into other characters or radicals. For example, “好” (good), a character with left-right structure, can be decomposed into “女” (daughter) and “子” (son). As illustrated in Figure 1(a), the left part of “好” is “女” and the right part is “子”. “女” and “子” are called the “radical” of “好”. Figure 1(b) is another example of the character “思” (miss) with an upper-lower structure.



(a) Left-Right Structure (b) Upper-Lower Structure

Figure 1: Examples of the structure of Chinese characters

One of the most important characteristics of character riddle lies in the structure of Chinese characters. Unlike the common riddles which imply the object in the riddle descriptions, character riddles pay more attention to structures such as combination of radicals and decomposition of characters. According to these characteristics, metaphors in the

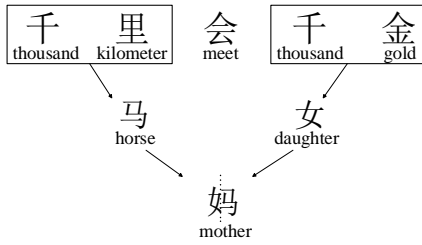


Figure 2: An example of Chinese character riddle: The solution “妈” is composed of the radical “女” derived from “千金” and “马” derived from “千里”.

riddles always imply the radicals of characters.

We show an example of a Chinese character riddle in Figure 2. The riddle description is “千里会千金” and the riddle solution is “妈”. In this example, “千里” (thousand kilometer) aligns with “马” (horse) because in Chinese culture it is said that a good horse can run thousands of kilometers per day. Furthermore, “千金” (thousand gold) aligns with “女” (daughter) because of the analogy that a daughter is very important in the family. The final solution “妈” is composed of these two metaphors because the radical “女” meets the radical “马”. Radicals can be derived not only from the meaning of metaphors, but also from the structure of characters. We will describe the alignments and rules in detail in Section 3.

In this paper, we propose a statistical framework to solve and generate Chinese character riddles. We show our pipeline in Figure 3. First, we learn the common alignments and the combination rules from large riddle-solution pairs which are mined from the Web. The alignments and rules are used to identify the metaphors in the riddles. Second, in the solving phase, we utilize a dynamic programming algorithm on the basis of the alignments and rules to figure out the candidate solutions. For the generating phase, we use a template-based method and a replacement-based method based on the decomposition of the character to generate the candidate riddles. Finally, we employ Ranking SVM to rank the candidates in both the solving and generation task. We conduct the evaluation on 2,000 riddles in the riddle solving task and 100 Chinese characters in the riddle generation task. Experimental results show that the proposed method outperforms baseline methods in the solving task. We also get very promising results in the generation task according to human judges.

2 Related Work

To the best of our knowledge, no previous work has studied on Chinese riddles. For other languages, there are a few approaches concentrated on solving English riddles. Pepicello and Green (1984) describe the various strategies incorporated in riddles. (De Palma and Weiner, 1992; Weiner and De Palma, 1993) use the knowledge representation system to solve English riddles that consist of a single sentence question followed by a single sentence answer. They propose to build the relation between the phonemic representation and their associated lexical concepts. Binsted and Ritchie (1994) implement a program JAPE which generates riddles from humour-independent lexical entries and evaluate the behaviour of the program by 120 children (Binsted et al., 1997). Olaosun and Faleye (2015) identify meaning construction strategies in selected English riddles in the web and account for the mental processes involved in their production, which shows that the meaning of a riddle is an imposed meaning that relates to the logical, experiential, linguistic, literary and intuitive judgments of the riddles. Besides, there are some studies in Yoruba (Akinyemí, 2015b; Akinyemí, 2015a; Magaji, 2014). All of these works focus on the semantic meaning, which is different from Chinese character riddles that focus on the structure of characters.

Another popular word game is Crossword Puzzles (CPs) that normally has the form of a square or rectangular grid of white and black shaded squares. The white squares on the border of the grid or adjacent to the black ones are associated with clues. Compared with our riddle task, the clues in the CPs are derived from each question where the radicals in solution are derived from the metaphors in the riddles. Proverb (Littman et al., 2002) is the first system for the automatic resolution of CPs. Ernandes et al. (2005) utilize a web-search module to find sensible candidates to questions expressed in natural language and get the final answer by ranking the candidates. And the rule-based module and the dictionary module are mentioned in his work. The tree kernel is used to rerank the candidates proposed by Barlacchi et al. (2014) for automatic resolution of crossword puzzles.

From another perspective, there are a few projects

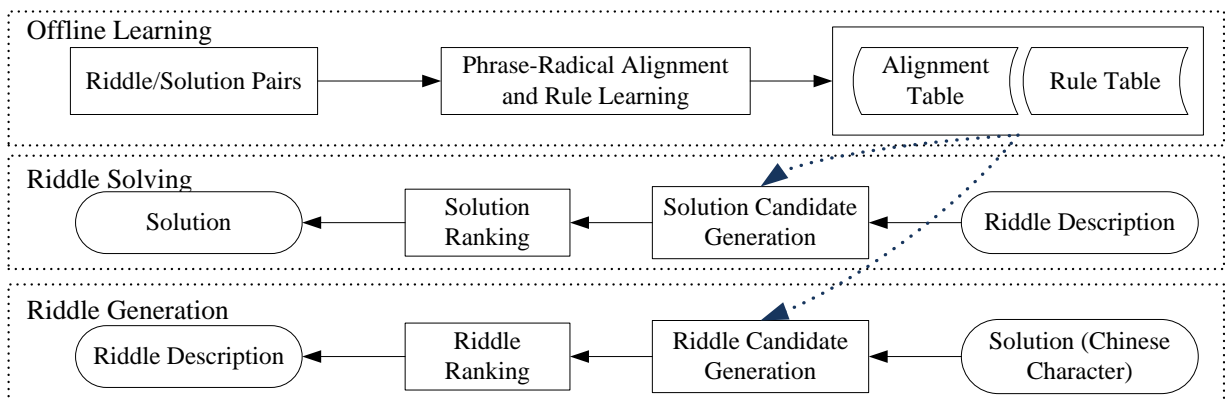


Figure 3: The pipeline of offline learning, riddle solving and riddle generation

on Chinese language cultures, such as the couplet generation and the poem generation. A statistical machine translation (SMT) framework is proposed to generate Chinese couplets and classic Chinese poetry (He et al., 2012; Zhou et al., 2009; Jiang and Zhou, 2008). Jiang and Zhou (2008) use a phrase-based SMT model with linguistic filters to generate Chinese couplets satisfied couplet constraints, using both human judgments and BLEU scores as the evaluation. Zhou et al. (2009) use the SMT model to generate quatrain with a human evaluation. He et al. (2012) generate Chinese poems with the given topic words by combining a statistical machine translation model with an ancient poetic phrase taxonomy. Following the approaches in SMT framework, it is valid to regard the metaphors with its radicals as the alignments. There are several works using neural network to generate Chinese poems (Zhang and Lapata, 2014; Yi et al., 2016). Due to the limited data and strict rules, it is hard to transfer to the riddle generation.

3 Phrase-Radical Alignments and Rules

The metaphor is one of the key components in both solving and generation. On the one hand we need to identify these metaphors since each of them aligns a radical in the final solution. On the other hand, we need to integrate these metaphors into the riddle descriptions to generate riddles. Thus, how to extract the metaphors of riddles becomes a big challenge in our task. Below we introduce our method to extract the metaphors based on the phrase-radical alignments and rules.

We exploit the phrase-radical alignments as to de-

scribe the simple metaphors, e.g. “千里” aligns “马”, which aligns the phrase and the radical by the meaning. We employ a statistical framework with a word alignment algorithm to automatically mine phrase-radical metaphors from riddle dataset. Considering the alignment is often represented as the matching between successive words in the riddle and a radical in the solution, we propose two methods specifically to extract alignments. The first method in according with (Och and Ney, 2003) is described as follows. With a riddle description q and corresponding solution s , we tokenize the input riddle q to character as (w_1, w_2, \dots, w_n) and decompose the solution s into radicals as (r_1, r_2, \dots, r_m) . We count all $([w_i, w_j], r_k)(i, j \in [1, n], k \in [1, m])$ as alignments. The second method takes into account more structural information of characters. Let (w_1, w_2) denote two successive characters in the riddle q . If w_1 is a radical of w_2 and the rest parts of w_2 as r appear in the solution q , we strongly support that $((w_1, w_2), r)$ is a alignment. It is identical if w_2 is a radical of w_1 . We count all alignments and filter out the alignments whose occurrence number is lower than 3. Some high-frequency alignments are shown in Table 1. For example, “四方”(square) aligns “口”(mouth) because of the similar shape and “二十载”(two decades) aligns “++”(grass) because “++” looks like two small “十”s.

Besides alignments are represented as common collocations, there is another kind of common metaphors concentrating on the structure of characters. We define 6 categories of rules shown in Table 2 to identify this kind of metaphors. A

Bigram Alignments	Radical	Frequency	Trigram Alignments	Radical	Frequency
西湖 (west lake)	氵 (water)	77	二十载 (two decades)	艹 (grass)	21
四方 (square)	口 (mouth)	40	党中央 (center of party)	口 (mouth)	19
千里 (thousand kilometer)	马 (horse)	36	意中人 (sweetheart)	日 (sun)	16

Table 1: The high-frequency alignments

Category	Description	Examples
Half	take half of the matched placeholder as radicals	[半折断边](.) [half,snap,break,side](.)
A-B	remove the B as radical in A to compose a new Chinese character	[减走无缺](.)(.) [subtract,leave,not,lack](.)(.)
UpperRemove	remove the upper-side radical of the matched placeholder	(.)[字]0,1[下南] (.)[character](0,1)[lower,south]
LowerRemove	remove the lower-side radical of the matched placeholder	[首前上北](.) [top,front,up,north](.)
LeftRemove	remove the left-side radical of the matched placeholder	(.)[字]0,1[右东] (.)[character](0,1)[right,east]
RightRemove	remove the right-side radical of the matched placeholder	(.)[字]0,1[左西] (.)[character](0,1)[left,west]

Table 2: The descriptions and examples of rules

rule is often represented as an operation that applies to a character for obtaining parts of it as radicals. For example, the character “上” (up) is usually represented as an operation to get the upper radical of the corresponding character. We extract the rules from the phrase-radical alignments we just obtain. In a phrase-radical alignment, if a radical appears in the one part of a character, we support that this radical is derived from this character, which means the other words in the phrase may describe an operation to this character. We replace this radical to a placeholder and generate a candidate rule with the corresponding direction by the radical position in this character. Thus, for each phrase-radical alignment $([w_1, w_n], r)$, we count $(w_1, \dots, w_{i-1}, (.), w_{i+1}, \dots, w_n)$ as a potential rule only if r is a radical of w_i . We count all rules learned from data, and filter out the rules whose occurrence number is lower than 5. Some rules are shown in Table 2. The word or phrase in the rule “A-B” mostly has the analogous meaning of “removing”. The word or phrase in the rule “Half” mostly has the analogous meaning of “half”. As for the rules “LeftRemove”, “RightRemove”, “UpperRemove” and “LowerRemove”, there are usually

a word or phrase that means “removing” as well as the others mean the “position” and “direction”.

We mine 14,090 phrase-radical alignments in total. More than 1,000 Chinese characters have at least one alignment, and there are 27 characters with more than 100 alignments. Common radicals are almost all contained in our alignments set. Chinese character is mostly composed of these common radical, so these alignments are enough for our task. We extract 193 rules in total for all categories of rules, all of them are applied to the riddle solving and the riddle generation.

4 Riddle Solving and Generation

4.1 Solving Chinese Character Riddles

The process of solving riddles has two components. First, we identify the metaphors in the riddle as much as possible by matching the phrase-radical alignments and rules, and integrate these metaphors to obtain a candidate set of solutions. Each candidate contains the corresponding parsing clues that imply how and why it is generated as its features. Second, we employ a ranking model to determine the best solution as output. Below we introduce our method to generate solution candidates, and we will

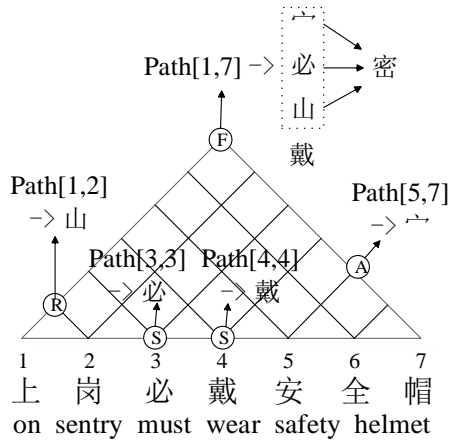


Figure 4: The decoding process of “上岗必戴安全帽”. -R: Path[1,2] records the clue that “上岗” matches “山” by the rule. -S: Path[3,3] records the clue that “必” matches itself and Path[4,4] records “戴”. -A: Path[5,7] records the clue that “安全帽” matches “宀” by the alignment. -F: We get a final solution candidate in Path[1,7] by above clues. In this example, the character “戴” from Path[4,4] is irrelevant to the solution.

introduce the ranking model in Section 4.3.

It is common that two metaphors do not share a character and the metaphor is composed of successive characters. Therefore, we utilize a dynamic programming algorithm based on the CYK algorithm (Kasami, 1965) to identify the metaphors with the help of the learned alignments and the predefined rules. We describe the algorithm in Algorithm 1.

An example to illustrate our algorithm is “上岗必戴安全帽”, where the corresponding solution is “密”. As shown in Figure 4, “上岗”(on sentry) aligns “山” by matching the rule “上(up)(.)” which means to take the upper part of the character “岗”. “必” and “戴” aligns itself. And the phrase “安全帽”(safety helmet) aligns to the radical “宀” by the alignments because of the analogical shape. Our ranking model will get the final solution “密” by these clues.

4.2 Generating Chinese Character Riddles

Two major components are required in the process of riddle generation. The first step is to generate a list of candidates of riddle descriptions for a Chinese character as the solution. The second step is to rank the candidate riddle descriptions and select the top-N (e.g. 10) candidates as the output. Below we

Algorithm 1: Candidate generation for riddle solving

Input : Riddle q , Alignment, Rule

Output: Path[1, n]

```

1 Tokenize the input riddle  $q$  to  $w_1, w_2, \dots, w_n$ ;
2 for  $len \leftarrow 0$  to  $n - 1$  do
3   for  $j - i = len$  do
4     if  $len = 0$  then
5       Character can align itself ;
6        $Path[i, j].Add([w_i, w_i] \rightarrow w_i)$  ;
7     end
8     else if  $[w_i, w_j]$  in Alignment then
9       Obtain the corresponding radical  $r$ 
10      in Alignment ;
11       $Path[i, j].Add([w_i, w_j] \rightarrow r)$  ;
12    end
13    else if  $[w_i, w_j]$  matches Rule then
14      Run the predefined operation of the
15      Rule, obtain radical  $r$  ;
16       $Path[i, j].Add([w_i, w_j] \rightarrow r)$  ;
17    end
18    foreach  $k$  in  $[i, j-1]$  do
19       $Path[i, j].Add(Path[i, k] \oplus$ 
20       $Path[k + 1, j])$  ;
21    end
22  end

```

introduce our method to generate candidates of riddle descriptions, and we will introduce the ranking model in Section 4.3.

We propose two strategies to generate the candidate riddle descriptions for a given Chinese character, called the template-based method and the replacement based-method, respectively. First we show our template-based method to generate riddles. The most natural method is to connect the metaphor of each radical. For a character and its possible splitting $RD = rd_i$, we select a corresponding metaphor by the alignment or rule, and then we connect all metaphor without any other conjunction words to form a riddle. The further method is to add a few conjunction words between each metaphor, which can make the riddle more coherent. We remove the recognized metaphors in riddle sentences,

Feature	Description
Correct_Radical	number of radicals matched
Missing_Radical	number of radicals not matched
Disappearing_Radical	number of radicals that disappear in all characters of riddle descriptions
Single_Matching	number of clues derived from character itself
Alignment_Matching	number of clues derived from alignments
Rule_Matching	number of clues derived from rules
Length_Rate	ratio of the length of clues
Frequency	prior probability of this character as a solution

Table 3: Features for riddle solving

Feature	Description
Riddle_Length	length in characters of the candidate riddle
Riddle_Relative_Length	abs(Riddle_Length-5) because the length of common riddles is between 3 and 7
Number_Radical	number of radicals that the character decompose
Avg_Freq_Character	average number of frequencies of characters in riddle
Max_Freq_Radical	maximized number of frequencies of characters in riddle
Number_Alignment	number of alignments used for generating the candidate
Length_Alignment	length of words from alignments
Number_Rule	number of rules used for generating the candidate
Length_Rule	length of words from rules
LM_Score_R	score of language model trained by Chinese riddles, poems and couplets
LM_Score_G	score of language model trained by web documents

Table 4: Features for riddle generation

and count the unigram and bigram word frequency of the rest words. These words are usually common conjunctions. We sample these words based on the frequency distribution and add them into the riddles to connect the metaphor of each radical.

Second, we use an alternative replacement-based method to generate the candidate riddle descriptions. Instead of generating the riddle descriptions totally from scratch, we try to replacement part of an existing riddle to generate a new riddle description. Let $w = (w_1, w_2, \dots, w_n)$ denote the word sequence of a riddle description on our dataset, where n denotes the length of the riddle in character. Let $[w_i, w_j]$ ($i, j \in [1, n]$) denote the word span that can be aligned to a radical rd , and let $X = (x_1, \dots, x_m)$ denotes the corresponding phrase descriptions of rd . We then replace $[w_i, w_j] \in X$ with the other alternative phrases descriptions of rd in X . We try all the possible replacements to generate riddle candidates. This method can generate candidate riddles that are more natural and fluent.

4.3 Ranking Model

Above we introduce the algorithm to solve and generate candidates, respectively. Then, we develop a

ranking model to determine the final output. Below we show the ranking model.

The ranking score is calculated as

$$Score(c) = \sum_{i=1}^m \lambda_i * g_i(c) \quad (1)$$

where c represents a candidate, $g_i(c)$ represents the i -th feature in the ranking model, m represents the number of features in total, and λ_i represents the weight of the feature. The features of riddle solving and riddle generation are in Table 3 and Table 4, respectively. We use Ranking SVM (Joachims, 2006)¹ to do the model training to get the feature weights. The weights of the features are trained with riddle-solution pairs. Specifically, in the riddle solving task, for the set of solution candidates, we hold that the original solution as the positive sample and others are the negative samples. Using the dynamic programming algorithm to obtain a list of solution candidates, the training process try to optimize the feature weights so that the ranking score of the original solution is greater than any of the ones from the

¹https://www.cs.cornell.edu/people/tj/svm_light/svm_rank.html

candidate list. In the riddle generation task, we select 100 characters on the basis of the frequency distribution of characters as a solution. For each character we use the riddle generation module to generate a list of riddle candidates. And we label these candidates manually where the better riddle descriptions get the higher score. Then the training process optimizes the feature weights.

5 Experimental Study

5.1 Dataset

We crawl 77,308 character riddles including riddle descriptions with its solution from the Web. All of these riddle-solution pairs concentrate on the structure of characters.

A stroke table, that contains 3,755 characters encoded in the first level of GB2312-80, is provided to describe how a Chinese character is decomposed into its corresponding radicals. Characters may have more than one splitting forms and a character is typically composed of no more than 3 radicals.

The data for training language model in riddle style include two parts: One is the corpus of riddles mentioned above, and the other is a corpus of Chinese poem and Chinese couplets because of the similar language style. We follow the method that proposed by (He et al., 2012; Zhou et al., 2009), to download the <Tang Poems>, <Song Poems>, <Ming Poems>, <Qing Poems>, <Tai Poems> from the Internet, and use the method proposed by Fan et al. (2007) to recursively mine those data with the help of some seed poems and couplets. It amounts to more than 3,500,000 sentences and 670,000 couplets. Besides the language model trained in riddle style, we also train a general language model with the web documents.

5.2 Evaluation on Riddle Solving

We randomly select 2,000 riddles from the riddle dataset as the test data, and 500 riddles as the development data, while the rest as training data.

Our system always returns a ranking list of candidate solutions, so we use the Acc@k (k = 1, 5, 10) as the evaluation metric. The Acc@k is the fraction of questions which obtain correct answers in their top-k results.

Giza++ (Och, 2001) is a common tool to extract

Feature Set	Acc@1	Acc@5	Acc@10
G	10.3	12.0	13.6
G+A	17.0	19.2	19.9
A	18.7	22.7	24.2
G+A+R	28.4	31.0	31.4
A+R	28.8	31.8	32.1

Table 5: Results of evaluation on test dataset with 2,000 riddles. -G: The alignments from GIZA++. -A: The alignments extracted following our method in Section 3. -R: Using the rules to identify the metaphors between the phrase and the radical following our method in Section 3. Our method (A+R) achieves better performances than the baseline methods from GIZA++.

Ranking Method	Acc@1	Acc@5	Acc@10
Jaccard Similarity	26.2	30.2	31.2
Ranking SVM	28.8	31.8	32.1

Table 6: Results of evaluation between ranking methods using the feature set (A+R). The Ranking SVM achieves better performances than the baseline metric from Jaccard similarity coefficient.

the alignment between bilingual corpuses. We use it as our baseline system that extracts the alignments automatically. And we use the Jaccard similarity coefficient as the baseline ranking metric. The Jaccard similarity coefficient is defined as:

$$J(A, B) = \frac{A \cap B}{A \cup B} \quad (2)$$

where A means the radicals set of the solution and B means the radicals set of the candidate.

The results are reported in the Table 5 and Table 6. The baseline method can only give about one-tenth correct solution at the Acc@1. Compared with the baseline model, by using the alignments extracted by our method, the system can improve 6.7% at the Acc@1 and 6.3% at Acc@10. A phenomenon is that only using the alignments we extract has the better results than combining it with the alignments from Giza++ because metaphors matching between phrases and characters are particular in our riddle task. Small changes in the phrase can affect the character that it implies and it may be not a metaphor even if a character in phrase is changed. Furthermore, by using rules to identify the metaphors in riddles, we get an improvement of 10.1% at Acc@1, which proves the validity of the

Score	Criterion
5	Elegant metaphors, totally coherent
4	Correct metaphors, mostly coherent
3	Acceptable metaphors, more or less coherent
2	Tolerable metaphors, little coherent
1	Wrong metaphors, incoherent

Table 7: The criterion of riddle evaluation

rule we define. The results prove that it is valid to use the alignments and rules that we extract to identify the metaphors in our character riddle task. The comparison between Jaccard similarity coefficient and our Ranking SVM method shows that the Ranking SVM is better with an improvement of 2.6% at Acc@1, which prove that compared to the Jaccard similarity coefficient, the Ranking SVM determine the solution more correct if we successfully identify all metaphors in riddle descriptions. Moreover, there is less improvement beyond Acc@5, which means the ranking model gets better results even if the system cannot identify all metaphors in riddle descriptions. We think that unlike the Jaccard similarity coefficient which only uses the features between the candidate character and the correct solution, the ranking model uses extra features in the riddles descriptions, e.g. the number of disappearing radicals, which helps to exclude obvious wrong candidates.

5.3 Evaluation on Riddle Generation

Because there is no previous work about Chinese riddle generation, in order to prove its soundness, we conduct human evaluations on this task in accordance with the following two reasons. Firstly, the generated riddles, which is different from the certain and unique solution in the riddle solving task, are varied. So it is hard to measure the quality of generated riddles with a well defined answer set. Secondly, small differences in riddles have a great effect on the corresponding solution. It may imply distinct radicals even if only a character in the metaphors is changed. The existing metrics such as BLEU, are not suitable for our task. Based on above analysis, each riddle that the system generates is evaluated by human annotators according to a 5 division criterion described in Table 7.

We randomly sample 100 characters following the distribution of the character as a solution. The

Method	Avg(Score)
Template-based Method	3.49
Replacement-based Method	4.14
Riddle from dataset	4.38

Table 8: Human evaluation of different methods

system generates riddle descriptions following the methods in Section 4.2 for each character. Sometimes the riddles we generate exist in our training data. We remove these riddles for the reason that we want to evaluate the ability of generating new riddles. In order to avoid the influence of annotators and compare the riddles generated by the system with the riddles written by human beings, the riddles are randomly disordered so that the annotators do not know the generating method of each riddle. For each character, we select 5 riddles generated by the template-base method, 5 riddles generated by the replacement-based method, and 2 riddles from the riddles dataset written by human beings, which form a set of 12 riddles in total. The annotators score each riddle according to the above criterion.

The result is shown in Table 8. The riddles written by human beings from the riddle dataset get the highest score than the riddles generated by the system. The riddles generated by the replacement-based method have a greater improvement than the basic template-based method. We consider that the replacement-based method retains some human information, which makes the generated riddles more coherent.

Another result is that the riddle whose solution is a common character or is composed of common radicals gets the higher score, which is explicit that we can get the better results if we have the more alternative metaphors of a radical.

Below we show two examples of the riddle descriptions generated with the solution “思”(miss) which often decompose into “田”(field) and “心”(heart) shown in Figure 1(b).

- 三星伴月似画里 (Three stars with the moon, like in the picture): The radical “田” is the inside part of “画”. The shape of “心” is three points and a curved line, which looks like three stars around a crescent.
- 日日相系在心头 (Every day in my heart):

The radical “田” is composed of two “日”s, and “心” occurs in the riddle description. The character “头”(top) means the radical “田” is on the top position.

6 Conclusion

We introduce a novel approach to solving and generating Chinese character riddles. We extract alignments and rules to capture the metaphors of phrases in riddle descriptions and radicals in the solution characters. In total, we obtain 14,090 alignments that imply the metaphors between phrases and radicals as well as 193 rules in 6 categories formed as regular expressions. To solve riddles, we utilize a dynamic programming algorithm to combine the identified metaphors based on the alignments and rules to obtain the candidate solutions. To generate riddles, we propose a template-based method and a replacement-based method to generate candidate riddle descriptions. We employ the Ranking SVM to rank the candidates on both the riddle solving and generation. Our method outperforms baseline methods in the solving task. We also get promising results in the generation task by human evaluation.

Acknowledgments

The first author and the fourth author are supported by the National Natural Science Foundation of China (Grant No. 61421003).

References

- Akíntúndé Akínyemí. 2015a. Riddles and metaphors: The creation of meaning. pages 37–87. Springer.
- Akíntúndé Akínyemí. 2015b. Yorùbá riddles in performance: Content and context. In *Orature and Yoruba Riddles*, pages 11–35. Springer.
- Gianni Barlacchi, Massimo Nicosia, and Alessandro Moschitti. 2014. Learning to rank answer candidates for automatic resolution of crossword puzzles. In *CoNLL*, pages 39–48.
- Kim Binsted and Graeme Ritchie. 1994. An implemented model of punning riddles. Technical report, University of Edinburgh, Department of Artificial Intelligence.
- Kim Binsted, Helen Pain, and Graeme Ritchie. 1997. Children’s evaluation of computer-generated punning riddles. *Pragmatics & Cognition*, 5(2):305–354.
- Paul De Palma and E Judith Weiner. 1992. Riddles: accessibility and knowledge representation. In *Proceedings of the 14th conference on Computational linguistics-Volume 4*, pages 1121–1125. Association for Computational Linguistics.
- Marco Ernandes, Giovanni Angelini, and Marco Gori. 2005. Webcrow: A web-based system for crossword solving. In *AAAI*, pages 1412–1417.
- Cong Fan, Long Jiang, Ming Zhou, and Shi-Long Wang. 2007. Mining collective pair data from the web. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 7, pages 3997–4002. IEEE.
- Jing He, Ming Zhou, and Long Jiang. 2012. Generating chinese classical poems with statistical machine translation models. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada*.
- Long Jiang and Ming Zhou. 2008. Generating chinese couplets using a statistical mt approach. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 377–384. Association for Computational Linguistics.
- Thorsten Joachims. 2006. Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226. ACM.
- Tadao Kasami. 1965. An efficient recognition and syntax analysis algorithm for context-free languages. Technical report, DTIC Document.
- Michael L Littman, Greg A Keim, and Noam Shazeer. 2002. A probabilistic approach to solving crossword puzzles. *Artificial Intelligence*, 134(1):23–55.
- Maryam Yusuf Magaji. 2014. Morphology, syntax and functions of the kilba folk riddles. *International Journal on Studies in English Language and Literature-IJSELL*.
- Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.
- Franz Josef Och. 2001. Training of statistical translation models.
- Ibrahim Esan Olaosun and James Oladunjoye Faleye. 2015. A cognitive semantic study of some english riddles and their answers in amidst a tangled web. *Asian Journal of Social Sciences & Humanities Vol, 4:2*.
- William J Pepicello and Thomas A Green. 1984. *Language of riddles: new perspectives*. The Ohio State University Press.
- E Judith Weiner and Paul De Palma. 1993. Some pragmatic features of lexical ambiguity and simple riddles. *Language & communication*, 13(3):183–193.

- Xiaoyuan Yi, Ruoyu Li, and Maosong Sun. 2016. Generating chinese classical poems with rnn encoder-decoder. *arXiv preprint arXiv:1604.01537*.
- Xingxing Zhang and Mirella Lapata. 2014. Chinese poetry generation with recurrent neural networks. In *EMNLP*, pages 670–680.
- Ming Zhou, Long Jiang, and Jing He. 2009. Generating chinese couplets and quatrain using a statistical approach. In *PACLIC*, pages 43–52.