Automatically Difficulty Grading Method of "Instruction System" Question Bank based on Knowledge Tree

Chengcheng Liu*
Nankai University, Tianjin, China
liuchengcheng600@126.com

Jin Zhang
Nankai University, Tianjin, China
nkzhangjin@nankai.edu.cn

Yang Zhou
Nankai University Tianjin, China
zhouyoung16@163.com

Shunjian Tian
tianshunjian1991@163.com

Xiaoli Gong
gongxiaoli@nankai.edu.cn

Xin Wei
sherry_0424@163.com

ABSTRACT

The aim of this study is to propose a model, which can automatically grade difficulty for a question from "Instruction System" question bank. The system mainly uses attributes which are employed to be input. A knowledge tree model which was established based on the proper nouns from Chinese "Instruction System" teaching material and a machine learning algorithm are utilized as important parts for classification. The experimental dataset comes from our built "Principles of Computer Organization" online education system, the accuracy result of difficulty classification could be 79.41% which is much higher than the accuracy of random guess 50%.

CCS CONCEPTS

•Computing methodologies → **Artificial intelligence; Natural language processing;** Machine learning; **•Applied computing** → *Education;* **•Information systems** → Data mining;

KEYWORDS

Automatically Difficulty Grading Model (ADGM)

ACM Reference format:
Chengcheng Liu, Jin Zhang, Yang Zhou, Shunjian Tian, Xiaoli Gong, and Xin Wei. 2017. Automatically Difficulty Grading Method of "Instruction System" Question Bank based on Knowledge Tree. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3076047

1 PROBLEM

The quality and quantity of question bank including User Generated Content(UGC) is very valuable, our laboratory launched the "Principles of Computer Organization" online education system whose website is "121.42.194.20". When achieving auto-generating test paper function module, the difficulty for the questions of a test paper need to be balanced. Then, how to automatically grade it for the multiple-choice Chinese questions that generated by different users from the system is a problem that need to be resolved.

*All of the authors come from College of Computer and Control Engineering, Nankai University, Tianjin 300-350, China

GECCO '17 Companion, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3067695.3076047

2 METHODOLOGY

2.1 Algorithm Architecture

Automatically Difficulty Grading Model (ADGM) is shown in Fig.1, for the first, a knowledge search tree model shown in Fig.2 (Each leaf node represents a knowledge point that was labeled by index with point and number) should be built according to a teaching material based on word segmentation module. Then, a useful dataset should be prepared for training the model, which should remove the oral words, stop words and duplicate words. For this problem, there are four kinds of attributes shown as Table 1 should be extracted. Finally, necessary features sorted out by certain methods act as input for machine learning, and the difficulty grading is the output.

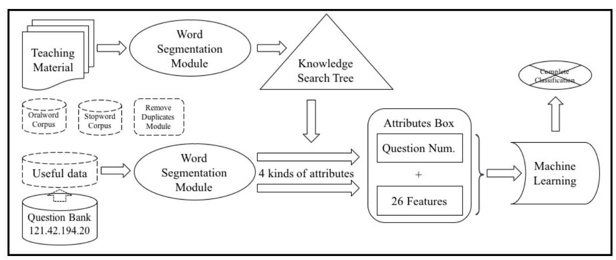

Figure 1: An overview of architecture

2.2 Attributes Box

2.2.1 Attribute of Knowledge Tree. Knowledge tree attribute contains of 5 sub-attributes. Hit knowledge node means a word segment of a question is the same as a leaf node of the knowledge tree. The larger the average span between hit knowledge nodes, the more difficult the question. As shown in Fig.3, what in the dashed square box represents the Index of the Common knowledge leaf Node (CNI) between two hit nodes. The span can be gained through "$Span(CNI, Index1) + Span(CNI, Index2)$, $Span(CNI, Index1) = \|Depth(CNI) - Depth(Index1)\|$".

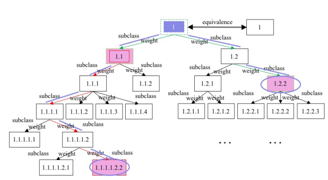

Figure 2: Knowledge Tree

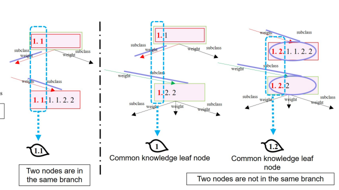

Figure 3: Common Node

Table 1: Attributes of question difficulty classification analysis

Knowledge Tree	Option Similarity
Knowledge Depth Knowledge Span Number of Knowledge Word Frequency Number of Relationship	Magnitude Knowledge Coverage Sentence Length Grammatical Morphology
Quantity Relationship	**User Portrait**
Word Number Analysis Length Non-Chinese Character No. Number of Unknown Words	Examination performance of question writers

2.2.2 Attribute of Option Similarity. Two options should be compared firstly, then compute the average. The option similarity (OS) between option X_1 and option X_2 can be computed as follows:

$$OS(X_1, X_2) = W * (SM, SK, SL, SC, SP)$$

W denotes the weight coefficient vector. "StringMagnitude=0" refers to the absolute logarithmic difference between 2 options>1, "1" means the other. StringKnowledgeSimilary refers to knowledge coverage similarity between 2 options. Firstly, to gain the option segmentation vector through the segmentation function `quecut()`. Then, to gain the vector with average depth, span, number, word frequency and number of unknown words /new words from the hit knowledge nodes through knowledge tree function `knowledgetree()`. Finally, compute Euclidean distance. The smaller the distance, the closer the options, the more difficult the question.

Algorithm 1 Similarity Value of SL,SC,SP

$Op1 \leftarrow Option1$; $Op2 \leftarrow Option2$

if len(Op1)==len(Op2)==0 **then**

$\quad StringLengthSimila = 1$

else

$$StringLengthSimila = 1 - \frac{\|(len(Op1) - len(Op2))\|}{(len(Op1) + len(Op2))}$$

$StringLengthSimila \in [0, 1]$

end if

RETURN StringLengthSimila

$StringCutLengthSimila$

$$= 1 - \frac{\|CutNo(Op1) - CutNo(Op2)\|}{CutNo(Op1) + CutNo(Op2)}$$

$StringCutLengthSimila \in [0, 1]$

RETURN StringCutLengthSimila

$StringPartSimila$

$$= \frac{GetPartSimila(Op1, Op2)}{MaxLength(CutNo(Op1), CutNo(Op2))}$$

$StringPartSimila \in [0, 1]$

RETURN StringPartSimila

3 EVALUATION

Experiment data, 1409 reliable questions have been cleaned from online education system, removing non-standard format, serious oral problems, superfluous spaces and so on. "Instruction System" knowledge tree shown as Fig.4 has been built by `Protégé` based on a textbook corpus cut by 24726 words and 1610 kinds of words.

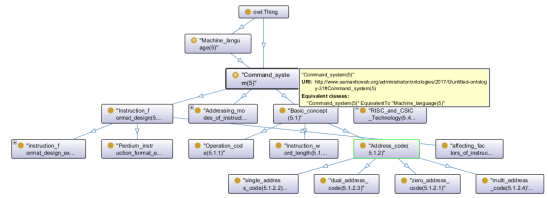

Figure 4: Sketch map of treetrump of knowledge tree

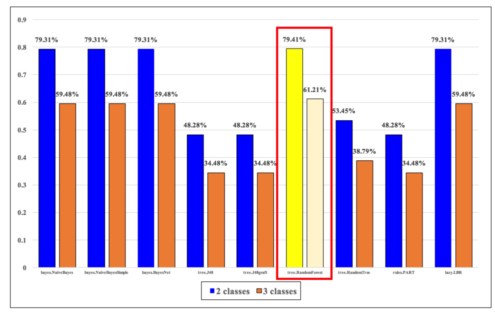

Figure 5: The accuracy of different algrothms

Table 2: Outcome Summary

Type	3 classes	2 classes
Correctly Classified Instances	71	270
Incorrectly Classified Instances	45	70
Kappa statistic	0.4157	0.5882
Mean absolute error	0.257	0.4189
Root mean squared error	0.46	0.4329
Relative absolute error	57.96%	83.78%
Root relative squared error	98.53%	86.57%
Coverage of cases (0.95 level)	75%	
Mean rel. region size (0.95 level)	46.26%	
Total Number of Instances	116	340

Table 3: Detailed accuracy

TP	FP	Precision	Recall	F	ROC Area	Class
0.7	0.184	0.667	0.7	0.683	0.818	1
0.512	0.28	0.5	0.512	0.506	0.677	2
0.629	0.123	0.688	0.629	0.657	0.857	3
0.612	0.2	0.614	0.612	0.612	0.78	W Avg.
0.882	0.294	0.750	0.882	0.811	0.885	1
0.706	0.118	0.857	0.706	0.774	0.885	2
0.794	0.206	0.804	0.794	0.793	0.885	W Avg.

Plainly, random forest is the best choice for the classifier as the results of Fig.5. Result of 3 classes by random forest is shown in Table 2, the correctly classified accuracy is 61.2069 % by using 66% percentage split for training data and 34% left for testing. The correctly accuracy of 2 classes is 79.4118% by using cross-validation with 10 folds. The detailed accuracy is shown in Table 3.

ACKNOWLEDGMENTS

Authors would like to thank Master Weilai Liu for providing the help of *ADGM*. Thank the anonymous referees for their valuable comments. The work is supported by awards under Grant No.CARCH201 604, No.20130031120028 and No.14JCQNJC00700.