AUTOMATED SHORT ANSWER GRADER USING FRIENDSHIP GRAPHS

Soumajit Adhya[1] and S.K. Setua[2]

[1]Department of Management, J.D. Birla Institute, Kolkata, India
smjt.adhya@gmail.com
[2]Dept. of Computer Science, University of Calcutta, Kolkata, India
sksetua@gmail.com

ABSTRACT

The paper proposes a method to assess short answer written by student using friendship matrix, representation of friendship graph. The Short Answer is a type of answer which is based on facts. These answers are quite different from long answers and Multiple Choice Question (MCQ) type answers. The friendship graph is a graph which is based on friendship condition i.e. the nodes have only one common neighbor. Friendship matrix is the matrix form of the friendship graph. The student answer is stored in a friendship matrix and the teacher answer is stored in another friendship matrix and both the matrices are compared. Based on the number of errors encountered from student answer an error marks is calculated and that number is subtracted from full marks to get student grade.

KEYWORDS

Friendship Graph, Friendship Matrix, Short Answer, Triplets

1. INTRODUCTION

Now-a-days various types of students' answers are evaluated by the examination systems. One is multiple choice questions answering which checks whether the student has marked the correct option or not. Another type is long answers which grades the student based on the content and writing style. The third type is the Short Answers (SA). SA are the type of answers, based on facts and concepts. Computer Assisted Assessment is common term for assessing student grades. This has become quite popular today because of the errors that humans make while evaluating an answer. To use computers for checking Multiple Choice Question (MCQ) type answers is very popular nowadays because of its speed and reliability. The development of Optical Mark Recognition (OMR) sheets have made it possible for examiners/examination system to use MCQ questions for conducting their examination. But MCQ questions do not test the true knowledge of the students. So there is need to check the student answers both short and long answers with the help of computer system. This paper proposes a system which can be used for marking SA by using computer system.

Generally there are 3 categories by which a SA can be marked. First, using statistical technique a SA can be marked. This is done by matching the student answer to the percentage of matching to the model answer by using various statistical techniques. Secondly using information extraction technique where the information is extracted from the text and those concepts are matched with the model answer. Third is the natural language technique which find the semantic meaning of

Natarajan Meghanathan et al. (Eds) : ACITY, VLSI, AIAA, CNDC - 2016
pp. 13–22, 2016. © CS & IT-CSCP 2016 DOI : 10.5121/csit.2016.60902

answer through parsing and finally compare it with instructors answer and assign the final scores[5].

This paper proposes a method by which the SA can be graded. It proposes a system which compare friendship matrices of given answers and model answers and accordingly provides the grade. It also provides a frame work by which the subject, predicate and object of each sentence is extracted teraining the semantic meaning of each sentence. So, this grading procedure takes into account semantic meaning of each and every statement. In this procedure the semantic meaning of the student answer is checked with semantic meaning of the model answer. This algorithm is a variation from C-Rater as it enters concepts from teacher answer one sentence at a time and stores it in a friendship matrix. Similarly the student answer is stored in another friendship matrix and is compared with the teacher answer to check how many concepts exactly matchs and then it is graded accordingly.

In this paper Section 2 deals with related work in automated SA grader, Section 3 deals with terminologies associated with this paper, Section 4 deals with the problem definition, Section 5 deals with the proposed method for SA grading and Section 6 is the Conclusion.

2. RELATED WORK

Normally automated SA grading can be graded into 5 eras.

Era 1 is the concept mapping technology where the following systems like ATM, C-Rater, Wang'08. The idea of concept mapping is to check whether a particular concept is present in the student answer or not and then grade accordingly. Concept mapping is based on the sentence level. Another technique is the facet mapping which is derived from concept mapping. Facets are the words and triples and any concept can be broken into facets.

The ATM (Automated Text Marker) breaks the student answer and teacher answer into concepts which are of few words and matches them to find the common concept and then it is assessed accordingly. The C Rater is aimed at to match the sentence level concepts between teacher and student answers. The two answers are compared by representing the texts using variation of syntax, anaphora, morphology, synonyms, and spelling correction. The teacher answers are entered as separate sentence for each every concept. Only one concept is assessed at a time. It gives a higher accuracy than other conceptual raters.

Era II is the Information Extraction technology where the following technologies like AutoMark, e max were developed. Since SA are usually expected to provide specific ideas and hence can be modeled by templates. Information extraction uses pattern matching, and also can extract structured data from unstructured sources and represent structured data as tuples.

Era III is using corpus based technology where the following technologies like Willow, Mohler-09 were developed. They use the statistical properties of large document corpora. Normally these methods are used for large texts but also can be effectively applied for SA also. But the correctness of student answer is limited only to the teacher answer vocabulary and can be increased by including synonyms and bilingual parallel corpora.

Era IV is the use of Machine Intelligence Learning technology where systems like e examiner and CAM was developed. These techniques uses the measurements extracted from natural language processing and combine them to form a single grade using classification or regression model.
Era V is the evolution era. This era is independent of methods. In this case the researchers all over the world compete in various competitions and tournaments [5].

3. TERMINOLOGY

3.1. Friendship Graph and Friendship Matrix

A graph is called a friendship graph if every pair of its nodes has exactly one common neighbor. This condition is called the friendship condition [2]. This graph is used to model the subject-predicate-object structure of a sentence. A friendship matrix is the relational or tabular structure of the friendship graph. In this case the common node (neighbor) is associated with the other nodes i.e. subjects and objects. The relation or table name is the common node and the subjects & objects are the nodes that are associated with it.

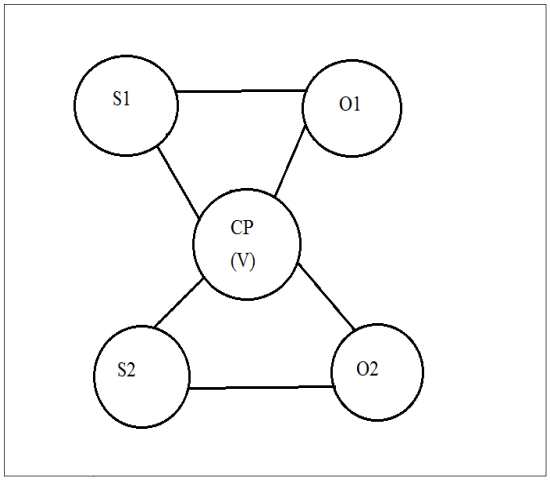

Figure 1: A friendship graph

Table Name: V (the common node)

SUBJECT	PREDICATE
S1	O1
S2	O2
S3	O3
S4	O4

Table 1: The Friendship Matrix which is derived from the Friendship Graph

The friendship graph structure is an ideal structure to store the RDF triplets. The common node can be used as the common MAIN PREDICATE, with the neighboring connected nodes as Subjects and Objects. Now this friendship graph structure can be saved in a matrix form by making the common MAIN PREDICATE as the TABLE NAME with PRE PREDICATE, SUBJECT and OBJECT as the column names.

PRE PREDICATE	SUBJECT		OBJECT		
	PRE SUBJECT	MAIN SUBJECT	PRE OBJECT	MAIN OBJECT	POST OBJECT

Table 2: A friendship matrix to represent the above friendship graph

Since the number of friendship matrix tables is going to be large so there will be a lot of overhead costs. So instead of maintaining individual friendship matrices we can save it in a single table with the following structure:

Common main Predicate	Corresponding Table					
Predicate1	Pre Predicate	Subject		Object		
		Pre Subject	Main Subject	Pre Object	Main Object	Post Object

Table 3: Final Friendship Matrix to store all the subject-predicate-object triplets derived from teacher student answer

3.2. Sentence Extraction

The sentence extraction algorithm extracts individual sentences from the document. We assume that the individual sentences are separated by full stops. By analyzing the full stops the sentence is extracted and each individual sentence is passed to the Co Reference Resolution pass one after another [4].

3.3. Natural Language Processing Techniques [4]

a. Co Reference Resolution (CRR) Pass: In this pass all the entities are recognized which are single words or a block of sequential words. Entities refer to any name, place etc. CRR attempts to find the words in a sentence that refers to an entity and replaces these references with the target entity. Then this modified sentence is passed to tokenization and parts of speech tagger.

b. Tokenization and Part of Speech Tagger: Each Sentence is tokenized and part of speech is tagged for each and every word. Then this tokenized sentence is sent to Full parsing phase.

c. Full parsing phase: In this case the sentence is written in Penn Tree Bank style which shows the phrasal structure and attachments. The nesting level is denoted by using tabs. Then this parse tree is sent for split coordinating conjunction phase.

d. Split Coordinating Conjunction Phase: Complex sentences are broken into simple sentences based on conjunction..

e. Extract Dependent Clauses: Sentences with dependent clauses, known as complex sentences in linguistics—as opposed to simple sentences with a single clause—are common in text. A dependent clause is introduced by either a subordinate conjunction

(for adverbial clauses) or a relative pronoun (for relative clauses), so those two cases have to be handled differently. This pass also extracts parenthesized phrases and clauses as they can be handled similarly, although not all are technically dependent clauses. Adverbial clauses are extracted into modifiers, whereas relative or parenthesized clauses are broken off into separate sentences.

f. Extract Adjective Phrases: Adjective phrases typically appear in sentences between one or two commas, and appear in the parse tree as nested under their subject.

g. Extract Prepositional Clauses: Prepositional Phrases are the main type of adjunct that is converted into a triple modifier. Because the attachments of modifiers are ignored by this system, attachments don't need to be captured.

h. Lemmatization: Reducing the verbs to their base form.

i. Synonym Conversion: All synonyms are checked from synonym table and converted into base word.

4. PROBLEM DEFINITION

SA do not have a huge length. There are no marks given for writing style for SA. This property distinguishes between SA and long answer. Today most of the commercial software only marks MCQ type questions. This will assess student's depth of knowledge only at lower level of Blooms taxonomy of educational objectives. They fail to assess student's performance at higher level of taxonomy of educational objective. So, writing SA for factual and conceptual answers have become a necessity now [5]. This paper proposes a method by which the SA can be graded. It also provides a frame work by which the subject, predicate and object of each sentence can be extracted so that the semantic meaning of each sentence is not lost. So, this grading procedure takes into account semantic meaning of each and every statement. In this procedure the semantic meaning of the student answer is checked with semantic meaning of the teacher answer.

5. PROPOSED ALGORITHM

This paper proposes a grader system for automatically marking SA. The student written factual answer is compared with the teacher answer. Both the answers which are written in paragraph forms are converted into friendship matrix form and then the two matrices are compared. Based on number of matches of tuples the student is given a grade.

Each sentence of teacher answer is extracted which is generally a complex sentence and is sent to NLP Converter. NLP converter converts the complex sentence which is extracted from the teacher answer into simple sentence(s). Then each simple sentence is passed to TTripletExtractor to create the Teacher Friendship Matrix.

Each sentence of student answer is extracted which is generally a complex sentence and is sent to NLP Converter. NLP converter converts the complex sentence, extracted from the student answer into simple sentence(s). Then each simple sentence is passed to STripletExtractor to create the Student Friendship Matrix.

The Grader will be applied to compare and to find the number of matches of tuples between the teacher friendship matrix and student friendship matrix. Based on number of matches the student

answer is graded. Every unmatched tuples or part of tuples of teacher friendship matrix and student friendship matrix is treated as errors. There are 4 types of errors i.e.

Error1: Error due to missing words in pre subject, pre object, and post object for matching subject/object

Error2: Error due to object of teacher answer not found in student answer for a matching main subject.

Error3: Error due to main subject of teacher answer not found in student answer for a matching common predicate.

Error4: Error due to predicate of teacher answer not found in student

To convert a complex sentence to simple sentence the following NLP techniques are used in order:

CRR, Tokenization and Parts of speech tagger, Full parsing, Split Coordinating conjunction, Extract Dependent Clauses, Extract Adjective Clauses, Extract Prepositional Clauses, lemmatization and Synonym Conversion[1][3][4].

The overall mehod is formalized as below:

Sentence Extraction (*Answer*)
{
Extract the sentences from model answer one by one.
}

NLP Converter (*A_Complex_Sentence)*
{
Use the existing NLP techniques to convert the complex sentences to simple sentences for each and every sentence of model answer.
}

TTripletExtractor (*A_Simple_Sentence)*

{

Step 1: Find the deepest verb from the Verb Phrase (VP) sub tree of the parse tree and match it in the predicate field. If the matching predicate is not found then add that predicate to the friendship matrix and go to Step 2. If the matching predicate is found then go to Step 2.

Step 2: While finding the deepest verb all the nodes that are encountered from the parse tree in the VP sub tree of the parse tree are combined to form a string and store it in the pre predicate field with corresponding to that common predicate which was found in Step 1.

Step 3: Find the first noun from the NP sub tree of the parse tree and store it in the main subject field with corresponding to that common predicate which was found in Step 1. While finding the first noun all the nodes that are encountered from the parse tree are combined to form a string and stored in the pre subject column.

Step 4: Find the first adjective, noun or pronoun from the VP sub tree of the parse tree and stored as object with corresponding to that common predicate which was found in Step 1.While finding the first noun/adjective/pronoun all the nodes that are encountered from the parse tree are

combined to form a string and stored in the pre object column and other nodes which followed object are to form a string and stored in the post object column.

}

STripletExtractor (*A_Simple_Sentence)*

{

Step 1: Find the deepest verb from the Verb Phrase (VP) sub tree of the parse tree and match it in the predicate field. If the matching predicate is not found then add that predicate to the friendship matrix and go to Step 2. If the matching predicate is found then go to Step 2.

Step 2: While finding the deepest verb all the nodes that are encountered from the parse tree in the VP sub tree of the parse tree are combined to form a string and store it in the pre predicate field with corresponding to that common predicate which was found in Step 1.

Step 3: Find the first noun from the NP sub tree of the parse tree and store it in the main subject field with corresponding to that common predicate which was found in Step 1. While finding the first noun all the nodes that are encountered from the parse tree are combined to form a string and stored in the pre subject column.

Step 4: Find the first adjective, noun or pronoun from the VP sub tree of the parse tree and stored as object with corresponding to that common predicate which was found in Step 1. While finding the first noun/adjective/pronoun all the nodes that are encountered from the parse tree are combined to form a string and stored in the pre object column and other nodes which followed object are to form a string and stored in the post object column.

}

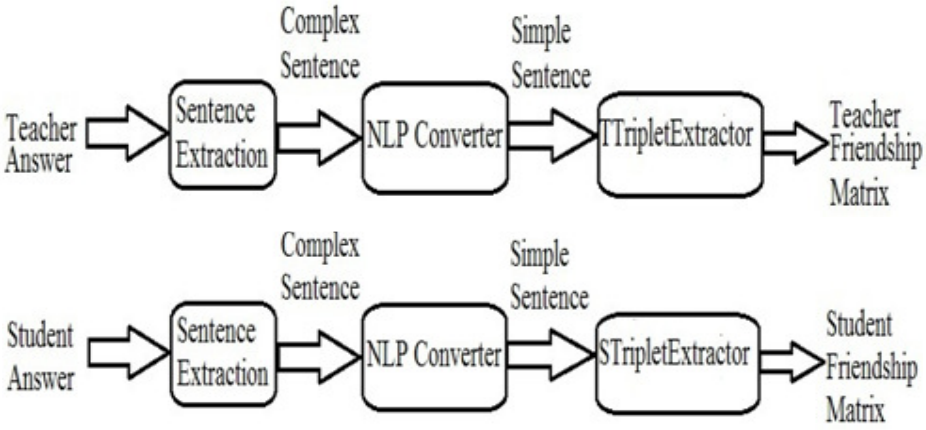

Figure 5: Process of Converting Teacher Answer and Student Answer into respective friendship matrix

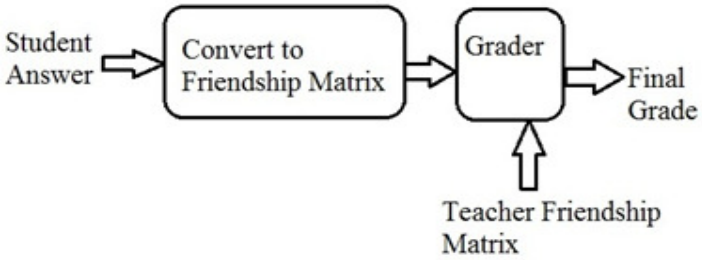

Figure 6: Grading the Student answers with help of teacher answers

Grader (*S_Friendship_Matrix, T_Friendship_Matrix)*
{
Step 1: Initialization of variables error_col_count, error_row_count, semantic_error and predicate_error.

Step 2: Take a common predicate from teacher friendship matrix and match with the common predicate of the student friendship matrix. If the common predicate is not found then go to Step 4 else go to Step 3.

Step 3: For each main subject for the matching common predicate from teacher friendship matrix repeat from Step 3.1 to Step 3.6

Step 3.1: Take a main subject from the teacher friendship matrix for the common matching predicate.

Step 3.2: Match it with the main subject of the student matrix for the corresponding matching predicate one at a time. If a match is found then go to Step 3.3 else go to Step 3.7

Step 3.3: Match the words present in the pre subject field to that corresponding matching main subject of the teacher friendship matrix with the pre subject field to that corresponding matching main subject of the student friendship matrix. For each unmatched word increase the error_col_count by 1 and go to Step 3.4

Step 3.4: Take the main object of the teacher friendship matrix to that corresponding main subject and match it with the main object of the student friendship matrix to that corresponding main subject. If match is found then go to Step 3.5 else increase the semantic_error by 1 and go to Step 3.7

Step 3.5: Match the words present in the pre object field to that corresponding matching main object of the teacher friendship matrix with the pre object field to that corresponding matching main object of the student friendship matrix. For each unmatched word increase the error_col_count by 1 and go to Step 3.6

Step 3.6: Match the words present in the post object field to that corresponding matching main object of the teacher friendship matrix with the post object field to that corresponding matching main object of the student friendship matrix. For each unmatched word increase the error_col_count by 1 and go to Step 3.1

Step 3.7: If no match is found increase the error_row_count by 1 and go to Step 2

Step 4: Increase the predicate_error count by 1. If all the common predicates are exhausted then Go to Step 5 else go to Step 2

Step 5: Calculation based on errors which are subtracted from the full marks i.e.

$$M = FM - EM$$

where,

$$MC = FM / N$$
$$ER1 = [MC/W]*ECC$$
$$ER2 = [MC/2]*SE$$
$$ER3 = MC*ERC$$
$$ER4 = MC*PE$$
$$EM = ER1 + ER2 + ER3 + ER4$$

Abbreviations:

M = Final marks of answer
FM = Full marks of an answer
EM = Total Error marks

MC = Marks of each concept
W = Average number of words present in pre subject, pre object, post object
SE = Semantic Error
ECC = Error Column Count
ERC = Error Row Count
ER1 = Total Error Marks occurred due to missing words in pre subject, pre object, post object for matching subject/object
ER2 = Total Error Marks occurred due to object of teacher answer not found in student answer for a matching main subject.
ER3 = Total Error Marks occurred due to main subject of teacher answer not found in student answer for a matching common predicate.
ER4 = Total Error Marks occurred due to predicate of teacher answer not found in student

}

6. CONCLUSIONS

This paper proposes a technique which uses facet mapping era. The NLP techniques is used to convert the text into base text from where the concept mapping technique is used to store the triplets in a friendship matrix, and to grade the answers. In SA concepts are checked rather than the writing style. In this case the concepts are compared between teacher answer and student answer and the number of errors is computed. These errors are then converted to a number which is subtracted from the full marks of the student answer. This technique can be expanded in future to mark long essays which depend on writing style. Also the future work relating this paper is to devise a faster information matching algorithm which can match student answer and teacher answer to grade them.

REFERENCES

[1] Delia Rusu, Lorand Dali, Blaž Fortuna, Marko Grobelnik & Dunja Mladenić, (2007) "Triplet Extraction from Sentences," Proceedings of the Conference on Data Mining and Data Warehouse (SiKDD 2007) held at 10th International Multi conference on Information Society

[2] Endre Boros, Vladimir A. Gurvich & Igor E. Zverovich, (2008) DIMACS Technical Report, 1 RUTCOR, Rutgers Center for Operations Research Rutgers, The State University of New Jersey

[3] Jonathan Hayes and Claudio Gutierrez, (2004) "Bipartite Graphs as Intermediate Model for RDF", The Semantic Web – ISWC 2004, Springer Berlin Heidelberg, Vol. 3298, pp 47 – 61.

[4] Aaron De Fazio, (2009) "Natural Question Answering over Triple Knowledge Bases", Australian National University

[5] Steven Burrows, Iryna Gurevych & Benno Stein, (2015) "The Eras and Trends of Automatic Short Answer Grading," International Journal of Artificial Intelligence in Education25, IOS Press, p. 60-117

AUTHORS

Soumajit Adhya holds a M.Sc degree in Computer and Information Science from University of Calcutta and currently employed as a IT faculty in JDBI, Department of Management.

S. K. Setua is an associate professor in the Department of Computer Science & Engineering at University of Calcutta. His research interest includes distributed computing, information & network security, big data analytics, SDN. He has more than 50 research publications in international journals and conferences.