Automated Essay Grading: An Application For Historical Malay Text

S.M.F.D Syed Mustapha[a] and N. Idris[b]

[a]*Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia. Tel: +603-79676301, E-mail: symalek@.fsktm.um.edu.my*

[b]*Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur 50603, Malaysia. Tel: +603-79676404, E-mail: norisma@fsktm.um.edu.my*

Abstract

Automated essay grading has been proposed for over thirty years. Only recently have practical implementations been constructed and tested. This paper investigated the role of the nearest-neighbour algorithm within the information retrieval as a way of grading the essay automatically called Automated Essay Grading System. It intended to offer teachers an individualized assistance in grading the student's essay. The system involved several processes, which are the indexing, the structuring of the model answer and the grade processing. The indexing process comprised the document indexing and query processing which are mainly used for representing the documents and the query. Structuring the model answer is actually preparing the marking scheme and the grade processing is the process of assessing the essay. To test the effectiveness of the developed algorithms, the algorithms are tested against the History text in Malay. The result showed that the information retrieval and the nearest-neighbour algorithm are practical combination that offer acceptable performance for grading the essay.

Keywords:

Information Retrieval; Nearest-Neighbour Algorithm; Indexing; Stemming algorithm; Grade Processing, Automated Essay Grading

Introduction

Teaching staffs around the world are faced with a problem on how to minimize the amount of time spent on the task associated with grading the students' essays. Grading essays will never be as simple as marking multiple-choice answers with the advent of large students numbers, the grading load has become time consuming. In addition, the subjective nature of essay assessment leads to variation in grades awarded by different human graders, which in turn is a source of unfairness. Due to these limitations, system that can automate the grading task is needed to eventually offer teachers individualized assistance in grading the student's essay.

In the West, researchers have been working on automation of student's essay for many years [1][2]. Even though there are arguments that automated systems can never model complexities of human grading, researchers continue to develop systems that can assess essays with relatively little effort and produces grades that generally close to the grading of human assessor. The earliest computer grading of essays was Project Essay Grade (PEG), developed by Page of Duke University [1][2][3]. The system was developed upon the idea that to extract surface features from the essays and then to compare these surface features with other essays using regression analysis. It focused on correlations between simple features of student texts and the grades assigned by teachers. It primarily relies on linguistic features of the essay documents. The second approach called Intelligent Essay Assessor (*IEA*), a software for assigning score to the quality of essay content

using the strategy-based approach called Latent Semantic Analysis (LSA) [1][4]. LSA methods concentrated on the conceptual content where the amount of content was counted by the concepts and not by the number of words. The third approach was known as Electronic Essay Rater (*E-Rater*) [5]. It uses a combination of statistical and natural language processing (NLP) techniques to extract linguistic features of essays to be graded [6].

In Malaysia, the research on the automated essay grading system is still lack of intention. However, from the survey, the idea of implementing the system is supported by the teachers. We discovered that grading student's essay for a course in any discipline presents a series of challenges especially for the factual text which involved a bag of words. Therefore, our main interest is on the History subject where it is a pure natural language which consist of significant words.

This paper describes our effort to provide an initial investigation of grading the essay automatically for the historical Malay text. In the second section, we reviewed briefly about the architecture of the system. In the third section, we described the process of the document indexing and query processing. Then, we described the process for structuring the model answer in the fourth section and explained the process of assessing the essay with the experiments in the fifth section. Finally, in the last section, we concluded with a discussion of the results.

The Architecture Of The System

Figure 1 showed the general view of the system where it contains major components of the system and shows the information flows between them. The system performed four major tasks which is query processing, document indexing, structuring the model answer and processing the grade. The information retrieval contributed techniques in the query processing, document indexing and also grade processing while the nearest-neighbour is being used in the algorithm for the grading process.

In summary, we suggested the following retrieval strategy:

- A sample of the document (answer scheme) was indexed.
- The indexed document was transformed into a model answer.
- The user specified the query as a natural language or free text query.
- The system converted this query into an indexed query.
- The indexed query was matched with the model answer in the grade processing.
- The system assigned grade to the indexed query.

Document Indexing And Query Processing

The life cycle of the information retrieval system begins with the development of the document representation. It is important to develop a text processing system which generate input text that is a plain text document into a document representative. Several processes which mainly used in the information retrieval system were adopted to this application in order to generate the document representative and to restructure the documents. The processes were composed into one main process called document indexing.

Indexing is an important process in an information retrieval system where it organizes text documents based on their contents [7]. It is a process to produce document identifiers which would be used to match with the identifier obtained from the queries. It is performed by assigning each document with keywords representing the document and the keywords must reflect the content of the document to allow effective keyword searching. The main purpose of the document indexing is to generate transactions whereby the words or the terms which are significant will be identified. Simple indexing is based on the words or word stems. There are several steps to perform the indexing process. This section describes briefly the steps in the indexing process.

Hyphens, Commas and Full Stops Removal

The hyphens, '-', found in the hyphenated words are usually discarded during the indexing process. This step also eliminates other special characters such as commas, ',' and full stops, '.'.

Separator Indication

After removing the full stops found in the sentences, a separator, '/', was put between the sentences. The function of these separators is to separate each sentence in the text.

Conversion of the Capitalized Words

All capitalized words are converted into small letters. Forcing the capitalized words to the lower case should be done before

the stemming process because only the lower case sequences are stemmed.

Stopwords Removal

Stopwords are those words that are frequently used and have little information value [8]. These stopwords are not searched for in the documents, and therefore ignored in the list of query terms. Stopwords are important because if a word appears many times in a document, it is less useful as a key to that document than the words that occur only a few times. Due to this situation, the stopwords are removed from the documents during the indexing process and consequently enhance the speed of the indexing process.

Stop-phrases Removal

Stop-phrase removal removes phrase in the text that does not contribute any information to the text or, sequence of words that provide no information about the text.

Words Stemming

One of the important techniques employed in the indexing process is the word stemming. Word stemming reduces the variant word form to a common form. By reducing the morphological variance of terms, we can improve the query-document matching process. In order to achieve this, the existing algorithm for stemming the Malay text adopted from [9] is modified and reinforced with few extra capabilities. The objective is to index documents based on the concept or word meanings in a specific context.

For the Malay morphology, there are more than one affix that can be attached to a word at the same time. There are several kinds of affixes which are suffix, prefix, infix and also prefix+suffix. However, the algorithm implemented only the most important rules from the two patterns of the rules which are the prefix and the suffix. The use of infix is very rare as people treat the resulting derived words as the root words. The prefix+suffix pattern was also omitted as we think that it is actually the combination of the prefix and the suffix rules. By using only two patterns of affixes that are prefix and suffix, we are not only reducing the numbers of the rule sets, but also the execution time for the stemming process. The algorithm is also capable of handling the spelling variations and exceptions in the root word when stripping of the prefix.

Features Recognition

There are some words in the text document which are significant, so-called keywords in the text document. Keywords are pre-selected term which can be used to refer to the content of a document. It can consist of words such as the names, places, nouns, numbers or certain patterns in the text files of documents. Feature recognition identifies the keywords in the text and assigned classes to the keywords. The keywords can be marked by putting the name of the class as the indexer or tagger besides each keyword in the document.

The document representative consists a list of recognizers and the keywords that are associated with each recognizer. Each recognizer represents a class of the keyword occurring in the text. A recognizer is assigned to a keyword if and only if one of its members occurs as a significant word in the text. These are referred to as the document's index term. The system will index the keywords by tagging the word with label that accord the class. Feature recognition for this application determined special features or keywords that to be indexed into several history classes such as *YEAR, PERSON, NAME, FACTOR, PLACE, STATES, COUNTRY, ACTION* and *EVENT* where the keywords will be classified.

Besides the document representation, there is a need to recognize the relationship between the structure of a query and the structure of a document. Thus, the structure of a query representation is also important and it is know as query processing. In order to provide the same structure as the indexed document, the query processing must follow the indexing process where many of the steps in the query processing are the same as those done in the document indexing. Figure 2 showed the flow of the indexing process where the natural language text is transformed into a structured text.

Structuring The Model Answer

Beside there is a need to have an effective technique for marking or grading an essay, a good design of the assessment also plays an essential role. Due to its importance to the system, the structure of the model answer needs to be structured properly. The original answer scheme can be used by the system as a source to build up the model answer. This section describes about how to prepare a model answer for the grading system.

The original essay of an answer scheme, which has been indexed by the document indexing process, is divided into several terms which contribute to the mark in an essay. These terms have several attributes and each attribute has its own values. These values are actually the list of keywords which are correlated to each other and formed the terms of the essay. In the model answer, there are several possibilities of answers associated with marks for each term in the essay. The highest

mark refers to the answers provided in the answer scheme while the others are the possible answers which might be given by the students. Therefore, to prepare those answers in the model answer, it is necessary to consider the possibility of all answers that the student may give. The marks attached to each answer are referred to the marking scheme provided by the teacher. A sample of the model answer generated by the system can be illustrated in Figure 3.

Grade Processing

This section offers an insight of the practice in processing a grade and the potential problems occurred. A simple nearest-neighbour learning algorithm that utilizes the Overlap metric is used to classify the class label for the user's query when given a training data [10]. The distance metric was devised to envision the relationships between the essay and the model answer. It determines the distance between the query and the marking scheme provided by the model answer by comparing the keywords of each attribute. If they are the same, then it returns a value of zero, otherwise a value of 1 is returned. The metric is often defined as follows:

$$\delta(x,y) = \begin{cases} 0 & \text{if } x = y \\ 1 & \text{if } x \neq y \end{cases}$$

Main Phases In Processing The Grade

The process for assessing the essay can be described as consisting of four main phases:

- Compare: The program must implicitly or explicitly compare the attributes of the user's query to the stored document or the model answer. In this case, each keywords for each attributes of the user's query is compared to the keywords of the model answer to gauge the similarity between them.
- Calculate: For each attribute, the program calculates the distances between the attribute's value of the query and the attribute's value in the model answer. Then, it sums the distances and assign the values to each instance in the training set.
- Retrieve: The overall distances for those instances would be the guidance for the program to make a decision about which instances to be retrieved. The program compares distances from all instances to retrieve the instances with the lowest distance. The retrieval of the instances is not based on the frequency of the occurrence of terms in the documents but mainly based on the most number of counts of the overlap terms between the query and the model answer. In other words, the retrieval of the instances is based on the frequency of the overlap terms between the query and the model answer.
- Categorize: The program makes a decision about which category or class should be assigned to the user's query. The class is assigned as a mark for each term in the essay.

These phases iterate until all the terms in the individual essay were checked and assessed. After all the processes completed, the program sums up all the marks and assigns the overall mark to the query. It determines the grade based on the rules of the marking scheme provided in the program. Finally, the program pass the grade to the essay to produce a graded essay. The full algorithm of the process can be viewed in Figure 4.

Experiments and Results

The experiments were conducted to test the algorithm for assessing or marking the student's essay using the historical data set. It was to determine the efficiency of the nearest-neighbour technique using the Overlap distance for calculating the distances of each attribute between the instances in the model answer and the query. From the experiment, it was found that the technique has shown a successful performance. The example of the results of the experiment is shown in Figure 5. Due to the results, the query was classified to *1* where the nearest neighbour of the query is the instance x_2 which has the class label *1*and the lowest distance value among the others. It retrieved the instance from the model answer and used the class label of the instance, which is the mark of that answer, to classify the query, x_q. Hence, the classification is correct and the class label is assigned to the query as a mark for that particular term.

The complete algorithm was also tested into a full data set which is available from the history text. The text has several terms to be assessed and the program examined the essay term by term. At the end of the process, the essay was graded by the process. However, the effectiveness of the grading process relies on how sufficient an information was fed into the model answer in order to provide a detail marking scheme for the grading process.

Conclusions

This paper represents an initial investigation of grading the essay automatically. It represented significant contributions by developing several techniques and algorithms such as

document and query indexing, structuring the model answer and processing the grade. It showed that the information retrieval and the nearest-neighbour algorithm are practical combination that offers acceptable performance for the domains like the historical Malay text. However, there are many directions in which this system could be enhanced. This will be our future work in order to produce a grading system which capable of giving the same result as the manual marking.

References

[1] Whittington, D., and Hunt, H. 1999. Approaches to the Computerized Assessment of Free Text Responses. *In Proceedings of the Third Annual Computer Assisted Assessment Conference*, pages 207-219.

[2] Wilson, D. R., and Martinez, T. R. 1997. Improved Heterogenous Distance Functions. *Journal of Artificial Intelligence Research*, 6:1-34.

[3] Page, E. B. 1994. Computer Grading of Student Prose Using Modern Concepts and Software. *Journal of Experimental Education*, 62(2):127-142.

[4] Foltz, P.; Laham, D.; and Landauer, T. K. 1998. The Intelligent Essay Assessor: Application to Educational technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning.*

[5] Burstein, J.; Kukich, K.; Wolf, S.; Chi, L.; and Chodorow, M. 1998. Computer Analysis of Essays. *In NCME Symposium on Automated Scoring.*

[6] Williams, R. 2001. Automated Essay Grading: An Evaluation of Four Conceptual Models. *In the Proceedings of the 10th Annual Teaching Learning Forum.*

[7] Adriani, M., and Croft, W. B. 1997. Retrieval Effectiveness of Various Indexing Techniques on Indonesian New Articles. Technical report.

[8] Callan, J. P.; Croft, W. B.; and Broglio, J. 1995. TREC and TIPSTER Experiments with INQUERY. *Information Processing and Management*, pages 327-343.

[9] Ahmad, F.; Yusoff, M.; and Sembok, T. M. T. 1996. Experiments with a Stemming Algorithm for Malay Words. *Journal of the American Society for Information Science*, 47(12):909-918.

[10] Payne, T. R. 1999. Dimensionality Reduction and Representation for Nearest Neighbour Learning. PHD thesis, University of Aberdeen.

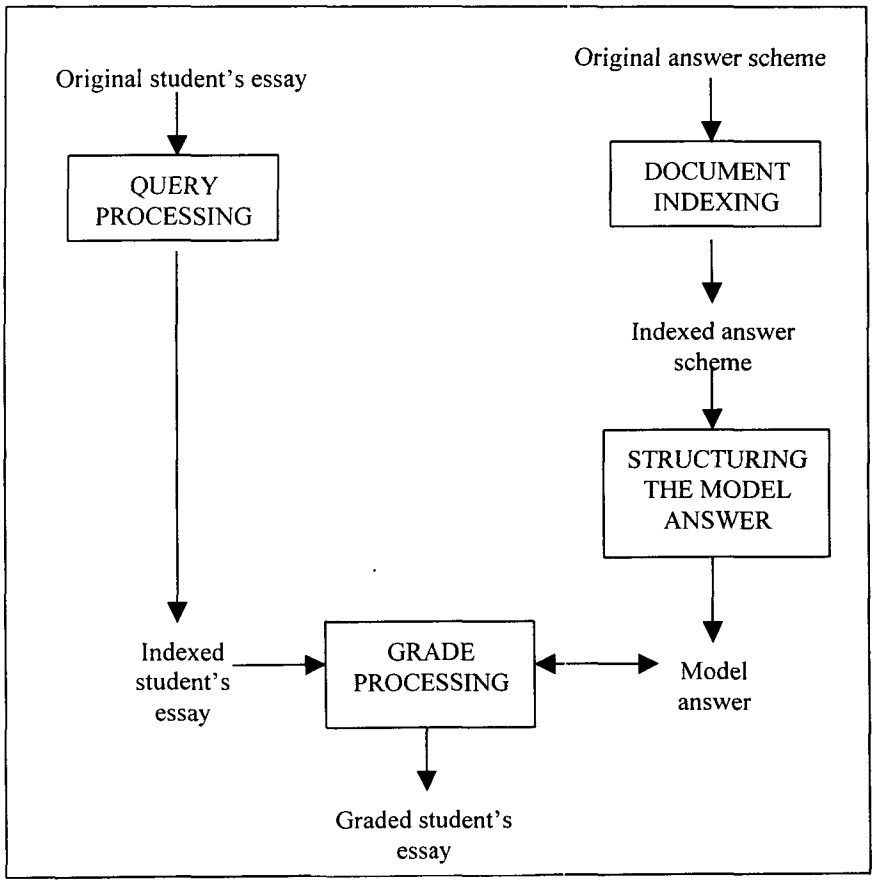

Figure 1: The architecture of the Automated Essay Grading System

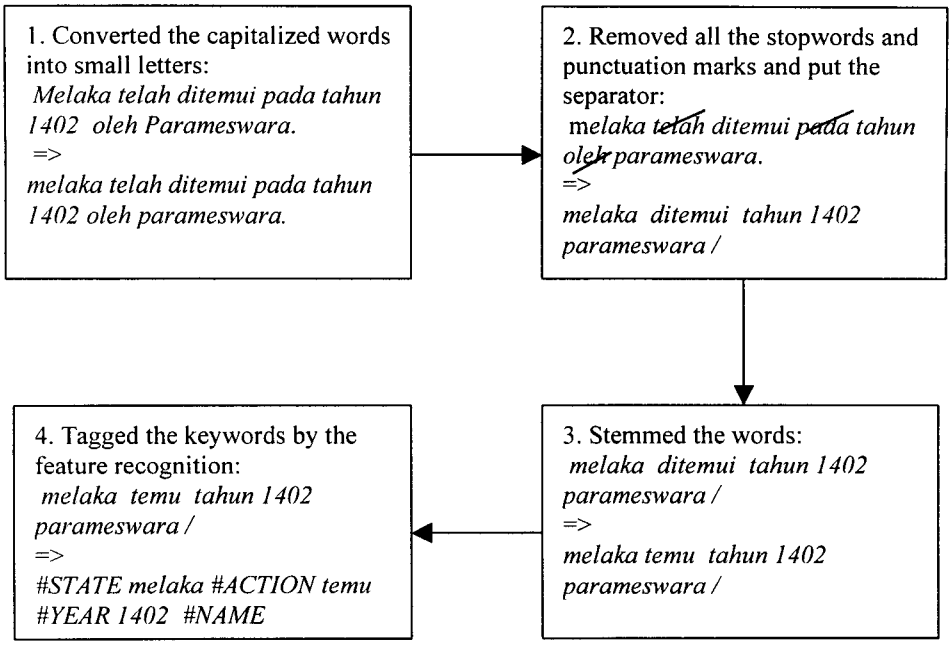

Figure 2: The flow of the indexing process

X_{11}	C	#ACTION	#FACTOR	#PERSON	#FACTOR	#PERSON
x_1	2	lahir	persatuan	guru	majalah	guru
x_2	2	tubuh	persatuan	guru	majalah	guru
x_3	1	lahir	others	others	majalah	guru
x_4	1	tubuh	persatuan	guru	others	others
x_6	0	lahir	others	others	others	others
x_7	0	tubuh	others	others	others	others

Figure 3: A sample of the model answer

Store the Model Answer
Read the Query (student's essay)
Read the Model Answer

While (more terms in the individual essay)
read query
read model answer
while (more instances in the model answer)

Calculate the distance:
while (more attributes)
read query (value of attribute)
read model answer (value of attribute)
Compare the values:
if (value(query)=value(model answer))
set d=0
else
set d=1

Sum all the distances:
$D=d_0 + d_1 + d_2 + \ldots\ldots + d_n$

Compare all the distances
Return instance with lowest distance
Assign the class value as the mark of the term

Calculate the sum of the marks:
$M = mT_0 + mT_1 + mT_2 + \ldots\ldots\ldots + mT_n$

Assign the sum of the marks to the student's essay
Determine the marking range to get the grade
Assign the grade to the essay

Figure 4: The algorithm for the grading process

X_{11}	C	#STATE	#ACTION	#YEAR	#NAME	d_0	d_1	d_2	d_3	D*i*
xq		melaka	temu	1402	tunperak					
x_0	2	melaka	temu	1402	parameswara	0	0	0	1	1
x_1	1	melaka	temu	others	parameswara	0	0	1	1	2
x_2	1	melaka	temu	1402	others	0	0	0	0	0
x_3	0	melaka	temu	others	others	0	0	1	0	1
x_4	0	others	others	1402	parameswara	1	1	0	1	2
x_5	0	melaka	others	others	others	0	1	1	0	2
x_6	0	others	others	others	others	1	1	1	0	3

Figure 5: Example of the result for calculating the distances between the instances and the query for each attribute in the model answer. The last column to the right of the table corresponding to the sum of the distances.