*IOSR Journal of Computer Engineering (IOSR-JCE)*
*e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 6, Ver. III (Nov. - Dec. 2016), PP 05-13*
*www.iosrjournals.org*

# Using Pattern Matching To Minimize Manual Processes in Assessment Administration

`Kavindu Augustine Mbii[1], Dr. Agnes Mindila, Phd[2],
Prof. Waweru Mwangi, Phd[3]
[2, 4]*Institute of Computer Science and Information Technology*
[1, 5] *Jomo Kenyatta University ofAgriculture and Technology,*

***Abstract:*** *Both the Kenya National ICT master plan 2014 – 2018 and vision 2030 placed ICT, Education and Training sectors as key pillars to attaining economic and social stability. Sessional paper No. 1 of 2005 stated that Information and Communication Technology had a direct role to play in education and if appropriately used, ICT could bring many benefits to the education and training sector in general. This research seeks touse ICTs to reduce manual processes in assessment administration by introducing an agent that matches an answer provided by the trainee in a question to phrases of patterns provided from the marking scheme. Further, the percentage matches produced are used to establish if the answer is correct or not. There is a very high possibility that if the process returns a big percentage match of over 60%, then the answer is correct. The concept under this paper uses NLP to solve the complexities that come with natural language. The WordNet module and Lema names properties are used to provide alternativesynonyms used in the marking scheme.This paper seeks to provide solution for the TVET education sector whose terminologies and answer expectations are more coupled and does not much depend on the language grammar.*

***Keywords:*** *Pattern Matching, Essay Questions Marking, NaturalLanguage Processing*

## I. Introduction

Assessments are essential tools of measuring achievements and progression in academics, but there havebeen challenges of implementing digital assessments while manual assessments register more disadvantages during theiradministration. The time and money spend on transporting assessment materials to and from the centers, ensuring security of the physical assessments papers while eliminating all risks and malpractices, employing all human resources required and the time spend on marking while trainees wait for results make the whole process costly and tiresome to the examiner and by a transfer of cost, to the examinee. Take for example the body mandated to administer assessments in Kenya, the Kenya National Examinations Council will require at least five months before they can release technical, vocational and entrepreneurship (TVET) results because of the processes involved.That's a long waiting time for the trainees who cannot continue with higher levels without the results.

The use of this pattern matching concept in assessment will be an ICT solution that would provide new dynamics in the TVET education and training sector while web based technologies could be employed to provide the implementing environment for these agent increasing its scope of implementation and reducing the assessment cost as well as making its administration more dynamic.

### 1.1 Pattern Matching Concept

Pattern matching or basically searching in artificial intelligence is the act of checking for the presence of the constituents of a given pattern in a given text where the pattern and the text are strings over some alphabet (Akinul 2014). Pattern matching is different from pattern recognition algorithms which aim at providing anequitableresponse for all inputs provided, taking into account their statistical variation while pattern matching algorithms look for exact matches with pre-existing patterns.

The simplest implementation concept in pattern matching would be comparing a pattern p against t, letter by letter until a match or end of text is found.

In a simpler explanation, take text X and text Y as the X and Y axis in a big graph and then, record a hit at every point where X and Y axis met or collude then out put the colluding words. The illustration below runs in O (m*n) since for each character in text t, matching is started with pattern p until mismatch occurs or until end of the text, so for large documents and lengthy patterns, this can be very slow. This has been solved by various existing algorithms like the Boyer Moore algorithm which has been chosen for this thesis because it has been a standard benchmark for the every practical string search theory. It starts searching and matching from the end of the needle, so it can usually jump ahead a whole needle length at each step hence reduced processing and matching time. It also uses the Index method which masks it a faster search algorithm based on the preprocessing of the text.

For example if we were searching for "001" in "010001" the process will be as shown below.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| i = 1<br>j = 0<br>**0**01<br>**0**10001 | i = 1<br>j = 1<br>**0**01<br>**0**10001 | i = 1<br>j = 3<br>001<br>010001 | i = 2<br>j = 0<br>**0**01<br>0**1**0001 | i = 2<br>j = 1<br>**00**1<br>0**10**001 | i = 2<br>j = 2<br>**00**1<br>0**100**01 | i = 2<br>j = 3<br>001<br>010001 | i = 3<br>j = 0<br>**0**01<br>0**1**0001 | i = 3<br>j = 1<br>**00**1<br>0**10**001 | i = 3<br>j = 2<br>**001**<br>0**100**01 |

Match was found at
i=3,
j = 2

**Figure 1.**1 Illustration of Basic Pattern Matching Concept

Pattern matching has been used in many application areas such as text editing, data retrieval, data filtering or data mining, DNA sequence matching, detecting suspicious keywords in security applications and many other applications.

Depending on the programing languages used, pattern matching can be used for function arguments, in case expressions. Itisoften possible to give alternative patterns that are tried one by one. This is achieved using tree patterns which can be used in programming languages as a general tool to process data based on its structure. A tree pattern is the part of a tree starting with the node that specifies some branches and nodes leaving some unspecified with a wildcard pattern. It helps in creation of the abstract syntax tree and algebraic data types.The earliest programming language with pattern matching constructs is SNOBOL which is an acronym standing for StriNg Oriented and symBOlic Language.SNOBOL4stands out from other programming languages in that it has patterns as data types whose values can be manipulated in all ways permitted in the programming language and that it provides operators for pattern concatenation and alternation (Eric Matthew Johnson, 1995)

**1.3using Boyer-Moore Searching Algorithm**

The searching algorithm in BM compares the symbols of the pattern from right to left. After a complete match the pattern is shifted according to its widest border. After a mismatch, the pattern is shifted by the maximum of the values given by the good-suffix and the bad-character heuristics.(R.S. Boyer, J.S. Moore, 1977)This is implemented with the simple text searching techniques discussed in 1.1 plus two heuristics which are occurrence and match. Occurrence refers to when the character in t [i+m-1] is not contained in the pattern p, then shift over to the position i=i+m while match refers to if a partial occurrence of t in p (Gesfield 1997). BM creates two tables, which are bmBc and bmGs making it easier to recognize the needed and unneeded characters in the pattern making the whole pattern matching process faster. After building a substring index, for example a tree or suffix array, the occurrences of a pattern can be found quickly. As an example, a suffix tree can be built in $\Theta(n)$ time, and all $z$ occurrences of a pattern can be found in $O(m)$ time under the assumption that the alphabet has a constant size and all inner nodes in the suffix tree know what leaves are underneath them ***(***Pekka 2005)**.** Table's*bmBc* and *bmGs* can be precomputed in time ***O*** (*m*+n-1) before the searching phase and require an extra-space in ***O*** (*m*n).

**1.3.1 Right-To-Left Scan and Character Shift**

This works in that, if pattern p is compared with text t and if the right most pattern symbol does not occur in the pattern at all, then the pattern can be shifted by *m* positions behind this text symbol. That is, check whether P occurs in T at some position in the right-to-left scanning manner.

**1.3.2 Bad Character Shift Rule**

This is applied if the text symbol that causes a mismatch occurs somewhere else in the pattern. Then the pattern is shifted so that it is aligned to this text symbol.

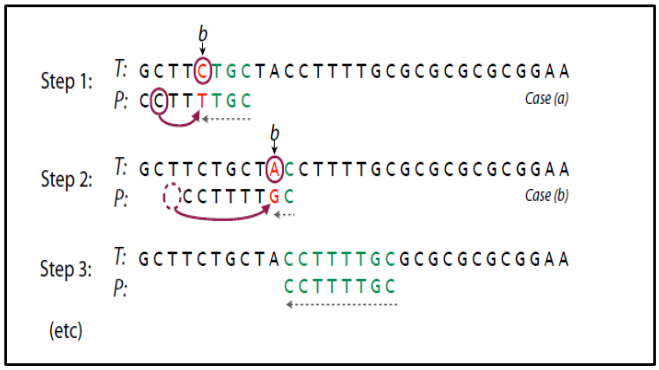


**Figure 1.3.2:** Illustration of Bad Character Shift

Realize that comparing first C-T causes a mismatch. Text symbol C occurs at positions 5,8,11 etc, in pattern p. p can then be shifted so that the next rightmost C after the mismatch in the pattern and aligned to text symbol C that caused the mismatch.

**1.3.3 Preprocessing for the Bad Character Heuristics**

For the bad character heuristics a function *occ* is required
Let *A* be the underlying alphabet.
The occurrence function *occ* : *A* A* is defined as follows:
Let *p A** with *p* = *p*0 ... *pm*-1 be the pattern and *a A* an alphabet symbol.
Then
*occ*(*p*, *a*) = max{ *j* | *pj* = *a*}
Here max( ) is set to -1.

**13.4 Good Suffix Rule**

Sometimes the bad character heuristics fail hence using the good suffix rule. For the good-suffix heuristics an array *s* is used. Each entry *s*[*b*] contains the shift distance of the pattern if a mismatch at position *b* – 1 occurs, i.e. if the suffix of the pattern starting at position bhas matched. In order to determine the shift distance, two cases have to be considered. A border is a substring which is both a proper prefix and a proper suffix of a string.

**Case 1**

In case 1, the matching suffix occurs somewhere else in the pattern, in which case it becomes better to derive the maximum possible shift distance from the structure of the pattern. This is computed as
If ***x _ = y***, then find the right-most copy ***t_*** of ***t*** in ***P*** such that ***t_*** is not a suffix of ***P*** and ***z _ = y***

In this case the matching suffix is a border to suffix of the pattern and have to be determined. It is also necessary that the border cannot be extended to the left by the same symbol, since this would cause another mismatch after shifting the pattern.

**Case 2**

This case is used when only a part of the matching suffix occurs at the beginning of the pattern. In this case the pattern is shifted by the longer of the two distances that are given by the bad character and the good suffix heuristics. That is
If ***t_*** does not exist, then find the largest prefix ***t__*** of ***P*** such that it is equal to a suffix of ***t***

**Case 3**

If***t_*** and ***t__*** do not exist, then p is taken to the end of the window.

The entire preprocessing algorithm of the Boyer-Moore consists of the bad character preprocessing and both parts of the good suffix preprocessing. It more efficient and faster on long patterns and improves the performance of pattern searching into a text by considering some observations and comparisons with other existing algorithms (Gesfield 1997).

**1.4natural Language Processing**

NLP is focused on developing computer systems that can communicate with humans using every day or natural language. A major goal in NLP is to produce systems which work with complete threads of discourse with human like abilities rather than only with isolated sentences (Russell & Norvig 2003). The results of research in the fieldof NLP are quickly put to wide applications in text categorization, automatic translation, topic extraction, and summarization. Automatic summarization involves reducing a text document into a short set of words that convey the main meaning of the text. (Bernard Marr, 2015).There are two types of automatic summarization. One is keyphrase extraction, its goal is to select individual words to tag a document, and the other is document summarization. It selects whole sentences to create a short paragraph summary. Coreferenceresolutionis the process of finding all expressions that refer to the same entity in a discourse. (Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, 2013). Modern systems have met this need for carefully designed features and global or entity-centric inference with machine learning approaches to coreference resolution. , is important for NLP tasks like summarization, question answering, and information extraction. Named Entity Recognizers allows easier application of techniques suited towards each task in NLP but a larger annotated data set is needed for it to effectively learn a model of named entities.

Discourse analysis is mostly driven by a binary tree built from non-overlapping text chunks. Elementary discourse units (EDUs) are segmented from the text, and serve as the leaves of the discourse tree. Many discourse parsers rely on a bottom-up approach for the tree building, linking EDUs and internal nodes with a parent relation level by level until encountering the single top root node(Jian Zhao, Fanny Chevalier, Christopher Collins, and Ravin Balakrishnan, 2012).Discourse parsing crosses sentence boundaries, extracting

relationships within an entire document. Discourse analysescan also be viewed as the semantic structure and its relationships within a text document. These discourse structures are the foundation of many text-based algorithms.(Vinu V Das, 2011).

There are three levels of natural language processing:

1. Syntax
2. Semantics
3. Pragmatics

Syntax is concerned with the proper ordering of words and its effect on meaning.

Semantics is concerned with the literal meaning of words, phrases, and sentences.

Pragmatics is concerned with the overall communicative and social context and its effect on interpretation.

Natural Language Processing and Information Retrieval are very different areas of study, and recently, very liitle effort has been put on investigating the use of NLP techniques for information retrieval. Simple Natural Language Processing techniques like stopwording and porter-style stemming have also been used in Information Retrieval and yield significant improvements, but higher-level processing like chunking, parsing and word sense disambiguation only yield very small improvements while increasing the processing and storage cost dramatically. Some Information Retreval related tasks, are question answering, information extraction, document clustering, filtering, new event detection, and link detection that can be combined with NLP, (Vinu V Das, 2011)

## II. Methodology

The sample questions were collected from three tertiary institutions. The researcher collected 47 question papers and 15 making schemes from Mwingi Teachers Training College, Pope John Paul Institute of professional studies, Kitui and University of Nairobi (Mwingi Center) as the target population. A target population is the total collection elements about which one wish to make some inferences (Mugenda and Mugenda 2003).

**Table 2.0: Target population**

| Study Group | Question papers | Marking schemes |
|---|---|---|
| Biological and Chemical sciences | 6 | 0 |
| Engineering and Information Technology | 9 | 10 |
| Communication and Educational courses | 12 | 1 |
| Business and Social studies | 20 | 4 |
| **Total** | **47** | **15** |

5 (five) question papers and their marking schemes from the Engineering and Information technology study group were used as the study sample. This was because the researcher had prior and advanced knowledge in the field of computing. The papers were in hard copy, so they were typed and saved as soft copies to serve as and input to the experimental phase of the study.

**2.1research Design**

The research used experimental design focusing on establishing the actual hypothesis and reaching tested conclusion of the relationships of various variables in the research. The researcher focused on matching an array of patterns: P = {p1, p2...,pk} tokenized using NLTK, which were simply strings of characters from a fixed pre created table T=t 1, t 2,...,tN . The aim was to find all occurrences of all the patterns of P in T and record the hits which was be used to compute percentages. With factors like Computer Literacy, readiness for Adaption, technical Support and government Regulations constant while hardware, software and network resources available, the marking scheme preprocessed, a sample of student answers were then provided for processing

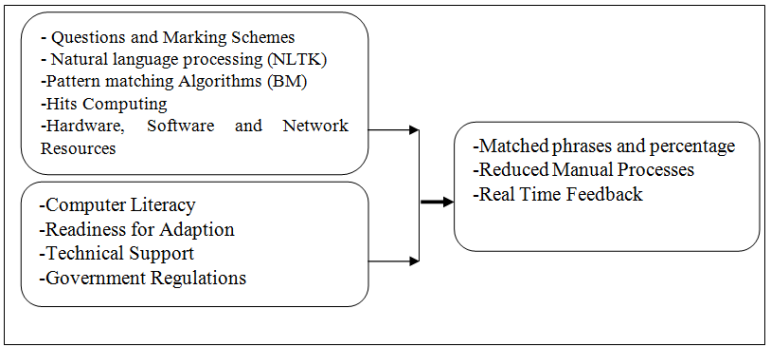


**Figure 2.1:** Theoretical Model

**2.2testing Enviromemt**

PYTHON waschosen for implementation of the testing environment. According to python 3.5 documentation, it has the largest standard library.Python has modules, exceptions, very high level dynamic data types, dynamic typing, and classes and combines remarkable power with very clear syntax. It runs on many UNIX variants, on Windows versions latter than 2000 and on the Mac as well. It is also an extension language for applications that need a programmable interface while it's portable and easy to learn.The researcher did not have prior programing background but the ease to learn of python programming enabled code writing from scratch.

The code is divided inmodules; the first module opens the marking scheme and answers, the second module tokenizes them, removes English stop words using NLTK and saves the output files in a text document. The third module is an iteration that gets all the posible synonyms of word that are in the tokenized answer, the fourth module has the matching code and the final module outputs the percentage match to the marking scheme using statistics.

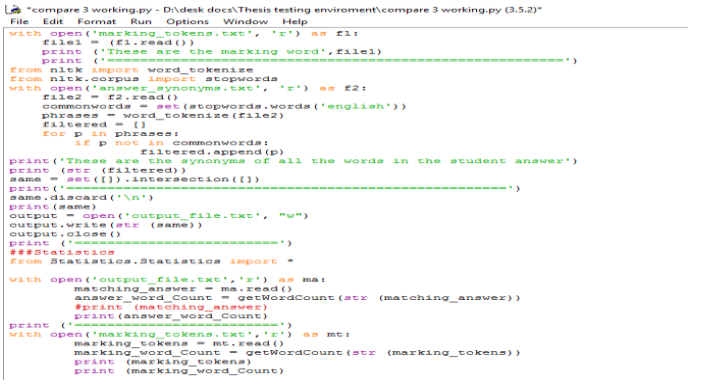

**Figure 2.2.0** Screenshot Of A Python Code Used In Testing

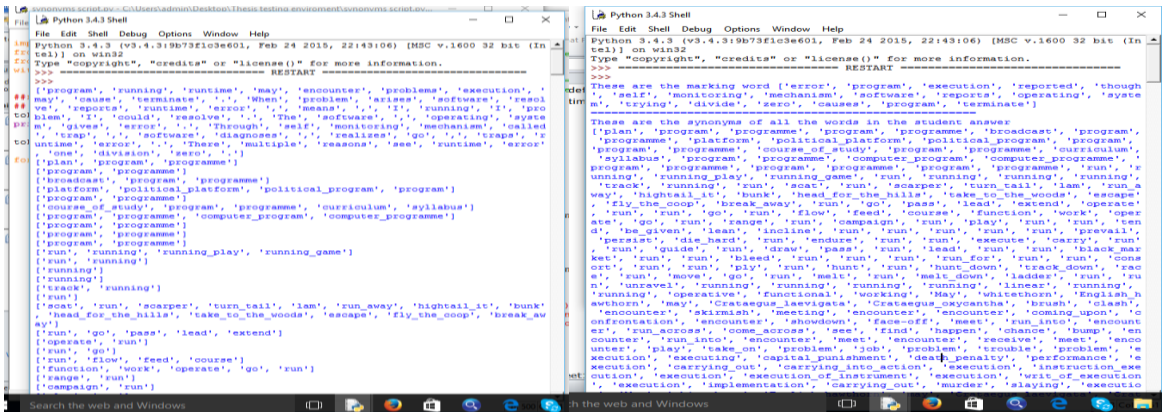

**Figure 2.2.1** Synonyms Modulescreen Shoot

The process starts with opening the .txt files of the student answer and the marking tokens. In python this is done by calling open method which is inbuild. The open files are futher tokenized by first importing a third party NLTK module and calling the .WordTokenize method. Once the tokenized documents are saved, the tokenized student answer is further opened the WordNet module is imported. Wordnet has a wide collection of English words. The lema names method is then called to print out all words with a similar meaning with all the words in the answer tokens. This is done word by word. A large document is produced with these synonyms which are compared to the marking tokens to print out every word that matches with the marking tokens as the reference token. This is achieved using the inbuild intersection method in python. The statistics module is a third party tool that has been used to computer the percentage match for determining if the answer is correct all wrong.

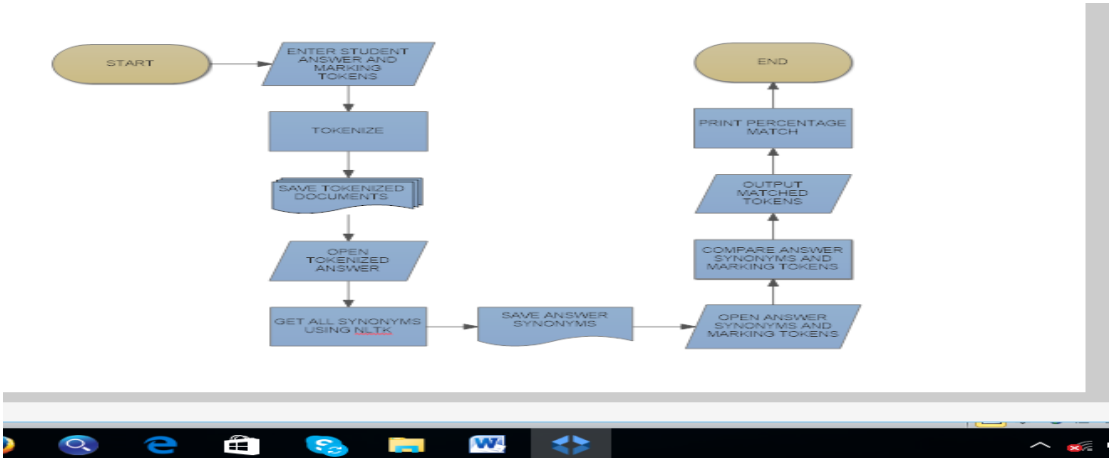


**Figure 2.2.2** Flow Chart Diagram

## III. Results

A total of six questions with correct answers and a small size of marking tokens, four correct answers and with a bulky marking tokens and two with wrong answers were processed

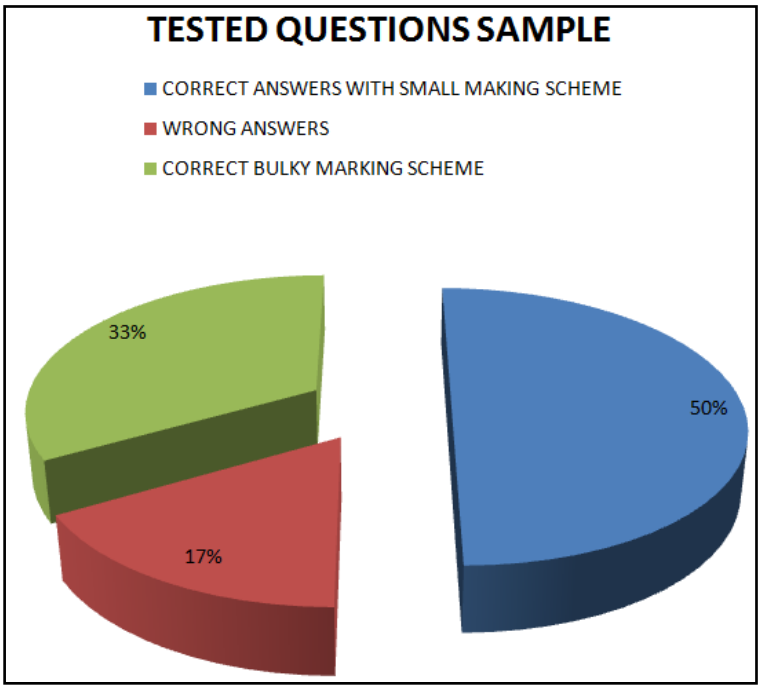


**Figure 3.0:** Tested Questions Sample

The tabulation of number of words per testing data is provides as the following

**Table 3.0** Tabulation of Testing Questions Data

| SNO | MARKING WORDS | STUDENT ANSWER | ANSWER SYNONYMS | MATCHED WORDS | PERCENT MATCH |
|---|---|---|---|---|---|
| 1 | 18 | 98 | 1254 | 11 | 61.11% |
| 2 | 9 | 198 | 2447 | 7 | 77.78% |
| 3 | 14 | 53 | 544 | 9 | 64.29% |
| 4 | 8 | 47 | 716 | 6 | 75.00% |
| 5 | 9 | 69 | 479 | 6 | 66.66% |
| 6 | 11 | 545 | 3032 | 8 | 72.73% |
| 7 | 212 | 545 | 3032 | 48 | 21.57% |
| 8 | 58 | 681 | 8143 | 20 | 34.54% |
| 9 | 44 | 75 | 628 | 9 | 20.45% |
| 10 | 75 | 181 | 1190 | 13 | 17.33% |
| 11 | 33 | 154 | 1825 | 1 | 3.03% |
| 12 | 28 | 60 | 388 | 1 | 3.57% |

## IV. Conclution

It was established that the marking tokens size affects the efficiency of the methods used. The higher the size of the marking tokens the less efficiency. From the figure 4.1 above, bulky marking schemes provides a lesser percentage of something between 15% and 35% even though the answers provided were correct. This is way below the anticipated 60% for the answer to be presumed correct.

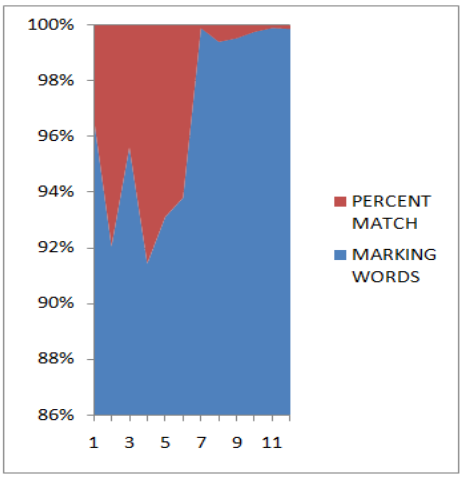


**Figure 4.1** Relationships between Marking Tokena and Percentage Match

It was however established that there was no relationship between the size of the answer the student provided and the percentage obtained after processing as shown in figure 4.2

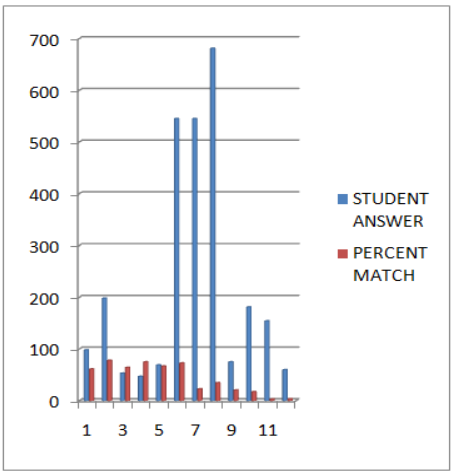


**Figure 4.2** Relationships between Answer Size and Percentage Match

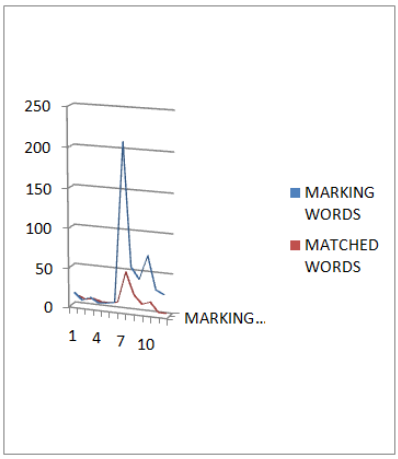


**Figure 4.3** Relationships between Marking Scheme and Matched Words

It was established that the matched words had a direct relationship to the marking scheme. The higher the matching words against the marking tokens, the better the match percentage obtained as shown of figure 4.3

Contrary to expectations that, there was no clear relationship between the answer provided by the student and the size of synonyms produced as shown of figure 4.4. It was found that this was the casebecause some words had more synonyms than others.

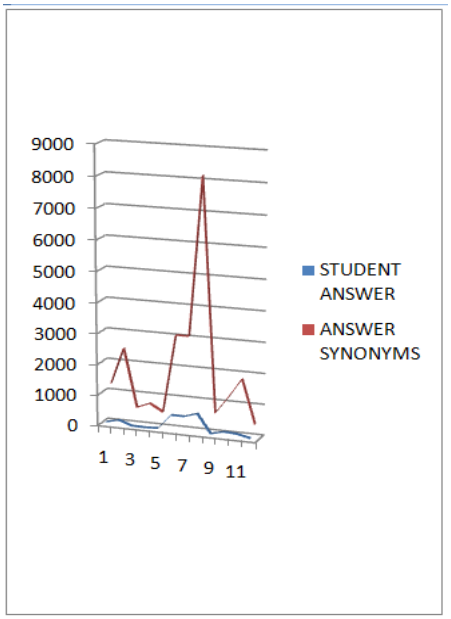


**Figure 4.4** Relationships between Student Answer and Answer Synomyns

### 4.1 The Nonsense Essay Theory

Les Perelman 2013, brought together nonsense essays that fooled software grading systems into giving high marks while criticizing the study by Shermis & Hammer (2012) about automated essay scoring (AES) of student essays as accurate as scoring by human readers. He provided different datasets in a close examination of the paper's methodology because of the widespread publicity surrounding the study.

The researcher'sopinion of the nonsense essay theory in relation to assessments on engineering and information technology domain would be that if a student submits a nonsense answer to the proposed agent and scores something above 60%, then the student must have had prior knowledge about the subject in question and deserves the percentages awarded. Terms and definitions in most scientific, mathematical, social and business domains are tightly coupled to their meanings and relationships in these fields.

## V. Recomendations

Pattern matching is a technic that can be used to grade student answers in more coupled subjects. This can be achieved with the help of natural language processing technics to get synonyms of words that the student will use in the answer. Natural Language tool kit is one of the many natural language tools that can be used for natural language processing. It offers a wide dictionary of English, Italian, Spanish and Greek words. More languages are still being processed.

## References

[1] Adam Retter, Web Applications and XML Technologies, Exist solutions, 2011
[2] Akinul Islam Jony, Analysis of Multiple String Pattern Matching Algorithms, Technical University of Munich, 2014.
[3] B. W. Watson and G. Zwaan. A taxonomy of sublinear multiple keyword pattern Matching algorithms. Science of Computer Programming, 27(2):85–118, 1996.
[4] Bell, J. Doing your research project. Buckingham, England: Open University. 1996
[5] Bernard Marr, Big data: using smart big data, analytics and metrics to make better decisions and improve performance, John Wiley and Sons, Inc., 2015
[6] Boyer R. S, Moore J. S, A fast string searching algorithm, Commun ACM, 1977
[7] Colazzo L, Silvestri L, The pragmatics of the Tutor: A proposal of modeling AI-ED97: 2009
[8] Eric Matthew Johnson, Computer programming for the humanities in SNOBOL4, Amazon, 1995
[9] Geoffrey C. Fox, Introduction to Web Technologies and Their Applications, Syracuse University, 1997
[10] Giraffa, Rosa, The use of Agents techniques on Intelligent Tutoring Systems, Congresso RIBIE, Brasilia 1998
[11] Gusfield Dan. Algorithms on strings, trees and sequences: computer science and computational biology. Cambridge University Press, 1997.
[12] Horspool, R. N. Practical fast searching in strings, Software Practice Experience, 1980

[13] Jennifer Robbins, Learning Web Design, IT systems, 2012
[14] Jian Zhao, Fanny Chevalier, Christopher Collins, and Ravin Balakrishnan, Facilitating Discourse Analysis with Interactive Visualization, IEEE, 2012.
[15] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky;Deterministic Coreference Resolution Based on Entity-Centric, Precision-Ranked Rules, Association for Computational Linguistics, 2013
[16] John A., Building Intelligent Agents, Bullinaria, 2005
[17] Lee R.C.T and Chin Lung Lu in Exact String Matching 2010
[18] Les C. Perelman, Critique of Mark D. Shermis & Ben Hammer, "Contrasting State-of-the-Art Automated Scoring of Essays: Analysis" -- Ver. 3.4 Mar. 13, 2013, Massachusetts Institute of Technology, 2013
[19] Martin Keen, Rafael Coutinho, Sylvi Lippmann, Salvatore Sollami, Sundaragopal, Venkatraman, Steve Baber, Henry Cui, Craig Fleming Developing Web Applications using Java Server Pages and Servlets, Redpaper, 2012.
[20] McKenzie S., Assessing quality of feedback in online marking databases: An opportunity for academic professional development or just Big Brother, Perth, (2004).
[21] Michael W. Berry, Jacob Kogan, Text Mining Applications and Theory, Wiley 2010
[22] Mugenda M.V, Mugenda A.G, Qualitative and quantitative approaches, Act press Nairobi; 1999
[23] National ICT master plan 2014 – 2018, MOICT, 2014
[24] Navarro G., Raffinot M., Flexible Pattern Matching in Strings, Practical Online Search Algorithms for Texts and Biological Sequences, Cambridge, UK. 2002.
[25] Nicholas Zakas, Professional JavaScript for Web Developers, Wiley 2012
[26] Nielson, J. Effective Use of Style Sheets, Jakob Nielson's, 1997
[27] Nikolaas N. Oosterhof, Philip K. F. Hölzenspies, and Jan Kuper. Application patterns. A presentation at Trends in Functional Programming, 2005
[28] Pekka Kilpel, Biosequence Algorithms, University of Kuopio, 2005
[29] Peter Bancroft, John Hynd, Jim Reye & Fabian Dal Santo: Web-based assessment submission and electronic marking, HERDSA 2008.
[30] Raita T., Tuning the Boyer-Moore-Horspool string-searching algorithm, SoftwarePractice Experience, 1992
[31] Richard Mansfield, CSS Web Design, Wiley Publishing, Inc. 2005
[32] Rotter, Kahn, & Anderson, Get Started with Cascading Style Sheets, Cnet 2000
[33] Russell, Stuart J. Norvig, Peter Artificial Intelligence: A Modern Approach (2nd ed.), Prentice Hall, 2003
[34] Sessional paper No. 1 of 2005: A Policy Framework for Education, Training and Research, Government Printer. 2005
[35] Stan Franklin, Art Graesser, taxonomy for Autonomous Agents, Springer Verlag, 1996
[36] Stenberg M, Artificial Intelligence - ELIZA and MegaHAL Analysis, G¨oteborg University, 2006
[37] Stoney G deGeyter, The Definitive Guide to Gathering, Sorting and Organizing Your Keywords into a High-Performance SEO Campaign, Pole Position Marketing, 2013
[38] Stuart Russell, Peter Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall, Inc. 1995
[39] Sunday D. M, A very fast substring search algorithm, Commun ACM, 1990
[40] Vision 2030 (2005) government press Kenya
[41] Vinu V Das, Computer networks and information technologies, Berlin Springer, 2011
[42] WATSON, B.W., 1995, Taxonomies and Toolkits of Regular Language Algorithms, Ph. D. Thesis, Eindhoven University of Technology, The Netherlands.
[43] Zhao, Jenson J. Web Design and Development for eBusiness. Prentice Hall. 2003