Research Article
Journal of Research and Review in Science, 174–183
Volume 4, December 2017
DOI:10.36108/jrrslasu/7102/40(0152)

JOURNAL OF RESEARCH AND REVIEW IN SCIENCE

ORIGINAL RESEARCH

PERFORMANCE EVALUATION OF ANAGRAM ORTHOGRPHIC SIMILARITY MEASURES

Raji-Lawal Hanat Y.[1,2], **Akinwale** Adio T.[2] ,**Folorunsho** O.[2], and Mustapha Amidu O.[2]

[1]Deparment of Computer, cience, Lagos State University, Ojo, Lagos

[2]Department of Computer, Science, Federal University Of Agriculture Abeokuta

Correspondence
Raji-Lawal Hanat Y., Department of Computer Science, Faculty of Science, Lagos State University Ojo, Lagos, Nigeria
Email:Halaw313@yahoo.com

Abstract:
Introduction: Anagram solving task involves a retrieving process from previously acquired knowledge, this serves as a suitable memory cognition test. Automation of this process can give a very good memory cognitive test tool, the method behind this automation is anagram orthographic similarity measure.
Aim: The purpose of this research is to study existing anagram orthographic similarity measures, to deduce their strengths and weaknesses, for further improvement.
Materials and Methods: Experiments were carried out on the measures, using real data. Their behaviour on different orthographic string set was observed.
Result: Experiment revealed that brute force has a very poor processing time, while sorting and neighbourhood frequency does not have issues with processing time.
Conclusion: The research revealed that existing anagram orthographic similarity measures are not suitable for character position verification and evaluation of syllabic complexity which are essential measures of working memory capacity.
Keywords: Cognition, orthography, similarity measures, bi-anagram, word permutation.

All co-authors agreed to have their names listed as authors.

1. INTRODUCTION

The word permutation has to do with the rearrangement of characters in a word to form other words. Several new words can be formed from a single word through word permutation. The rearrangement of the characters of a word to form another word of the same length n-gram but different position is called anagram. This is relevant for user's cognitive analysis [8], analysis of user's memory capacity, analysis of user's word fluency through anagram solving ability.

Verifying the memory stability of individual is mundane at certain levels. In cognitive deficit people due to Alzheimer's disease, during the recovery process or medical monitoring of patients, it is important that patients are observed psychologically from time to time. Anagramming is a very good measure of recovery level in this case. The psychological status can further be measured through the kind of anagram that can be solved by patients. [9,10]Ability of patients to generate hard or average or simple anagram can quantify their level of recovery in terms of cognitive ability.

While choosing career during teenage age, cognitive ability can depict memory smartness of individuals. Challenging courses like medical sciences, engineering, law e.t.c. requires candidates with high cognitive ability. This is to enhance and ease student's study of a course. Anagram can serve as a good measure of candidate's intellectual capacity. Thus, it can be used as one of the measures for taking decision during career counselling of teenagers. It gives a true picture of working memory capacity of students. Also, Anagram solving can be used to enhance student's spelling mastery ability. Anagrams also serve as a means of data encryption which is applicable to network security.[5,7,11]

This research explored anagram orthographic similarity measures, by considering their strengths and draw backs. This is to determine the necessary enhancement needed by these measures.

Anagram orthographic similarity measures were analysed, and thus serves as analysis measure for anagrams solved by users for various purposes. Through these measures deductions can be drawn on users based on their anagram solving ability.

1.1 Motivation

- It is necessary to devise a standard means of monitoring recovery rate and level of patients with cognition problem like aphasia disease through anagram autographic similarity measure. [9],[10]
- It is necessary to improvise for a cognitive career counsellor. This is due to the fact that, cognitive test reveals working memory capacity, and can in turn be used to predict student's ability to withstand some courses of study. One of the steps to this is performance evaluation of anagram autographic similarity measure, which will reveal the lapses of existing measures for enhancement. [8]
- Several applications had swept students attention from reading, which is a defect factor of student's reading fluently. Anagram solving task is fun fill and also educative in terms of word at sight and reading fluently for students. [5], [7], [11]

2. MATERIAL AND METHODS

The authors in [1] researched on the effects of syllabic structure and grapheme frequency of target words on an anagram solving task. He also checked for presumable differences on the solving performance of average and below average readers. This was achieved by measuring time spent and user's number of moves for solving different non-sense anagrams. Data analysis revealed that anagram solving time is affected by the syllabic structure of target words. The main effect of syllabic complexity indicates that both groups i.e. average and below average readers were equally affected. The effect of syllabic complexity was also noticed in the reading fluency measure, where the above average group out perform the below average group. The two experiment groups achieve lower time scores at the simple syllabic structure. The syllabic structures used in the research are: consonant-vowel(CV), consonant-consonant-vowel(CCV), consonant-vowel-vowel(CVV) and consonant-vowel-consonant(CVC). An anagram solving task can be conceptualized as a lexical access task. [2]

The authors in [3] investigated lexical decision task, to be able to deduce that "poor motion" detectors should encode letter position less accurately than good motion detectorss. He thus, inferred that "poor motion detector" should be more likely to unscramble briefly presented anagrams and respond to them as if they were words.

The authors in [4] introduced the use of anagram for resisting evasion by polymorphic mimicry attacks, randomization strategy was used to achieve this.

One of the tools used in psychology to investigate cognitive processes is anagrams. This research applied rasch scaling to a sample of five-letters anagrams to examine whether a single undimensional scale based on syllable number can be usefully applied. Rasch analysis was used to establish a participant's ability to solve anagrams, and also to establish the relative difficulty of each anagram. Rasch's theory suggests that the probability of getting an anagram correct is caused by the difference in a person's ability and the anagram's difficulty. It was deduced from the research that syllable number is a major factor in determining anagram difficulty. The more syllables the target word contain, the harder it was to solve and the higher the rasch score. Thus, syllable number is a compounding factor for other variables that contribute to anagram difficulty. This research made a useful contribution to measurement models of human cognitive problem solving [5].

The authors in [6] used high-density (64 channel) event related potential(ERP) combined with di-pole source analysis to explore the neural bases of unconscious error detection in an anagram solution task. ERP studies indicates error negativity (Ne/ERN) and error positivity(Pe), they are used for error monitoring and error detection. Responses were collected for unconscious error detection(UED), No error(NE) and detection error(DE), the response time was also recorded i.e. the time from when anagram appear and response. The research analysed the ERPs elicited by the Chinese anagrams, and deduced that amplitude of N2 i.e. negative ERP elicited by the UED response was greater than those of NE and DE responses. Late positive component elicited by the DE response was greater than those of the UED and NE responses.

Anagram priming effect can be found in lexical decision task, with non-adjacent transposed letters. Anagram relationship within lexicon or between stimuli, constitute a form of orthographic similarity whose effect on reading performance is detectable. orthographic similarity on anagram can be detected using neibourhood frequency effect, while anagram relationship acts as orthographic similarity factor [7].

Researchers in [7] worked on effect of anagram relationship between lexical units, prime and target stimuli, the research deduced that mental lexicon is activated by position-free letter codes, together with other units that encode the order information.

Robert and Debra in [8] worked on anagram software for cognitive research, the software provides different modes of operation: interactive and automatic. The interactive mode allows a researcher to provide a hard generated list of test strings or words for evaluation of possible anagram. The list can be entered either interactively or through an input file. The automatic mode generates a specific number of fixed length test string pseudo randomly. In either case, the test string is converted to a sorted order, and each possible sub-string is considered. All possible anagrams are identified, and the lemma frequency information for all orthographically identical word forms is summed and printed. The research did not consider bi-gram frequency in anagrams.

The processes involved in working through problems to solve anagram was reported in [9]. The authors looked into the role of effort in remembering anagrams and their solutions. Firstly, experiment was conducted to compare the effect on memory of copying word, typing them as solution for easy/hard anagrams, and solving anagrams. Secondly, memory for anagram using frequency judgement was compared. It was deduced that memory for difficult anagram is not better than memory for easy one.

Sandra et. al. in [10] researched on a new measure, the Northwestern Anagram Test (NAT), which was developed to test accuracy of word order (syntax) in sentence production in patients with speech production, word comprehension and/or word-finding difficulties, or reduced working memory capacity. The anagram method requires the assembly of individual word cards presented in scrambled order into meaningful sentences. A related test for aphasia, the verb and sentence test (VAST17), includes an anagram task; however, its major purpose is to evaluate verb and verb argument structure. Thus, the stimulus cards group words to form complete sentence constituents (e.g, complete noun phrases: The boy / is chasing / the dog.) on the other hand, the VAST examines only active and passive sentence structures. The NAT provides a separate stimulus card for each word of the target structures such that the patient has few clues as to permissible word combinations. In addition, like the NAVS, the NAT examines several canonical forms. The research report results obtained with the NAT from patients with primary progressive aphasia (PPA) and their cognitively intact controls. Northwestern Anagram Test performance was compared with performance on the sentence production primary test(SPPT) of the NAVS and was compared with measures of aphasia severity, naming, single word comprehension, oral apraxia, and object recognition to demonstrate its specificity for syntax. It was hypothesized that the NAT would correlate with measures of syntax but not with measures of naming, word comprehension, or motor speech production. Furthermore, it was predicted that performance would be influenced by grammatical complexity [10].

The canonical well known model of word recognition, which stem from Interactive Activation Model of Word and Letter Perception, use slot-filler representations for letters, and are thus poorly suited to capture letter permutation effects. Henin et. al. in [11] proposed a solution that takes up a prominent feature of the early interactive activation models: units corresponding to structures at multiple spatial (and temporal) scales (e.g. phoneme, bigram, word). Unlike in the canonical models, the units in this model detect structure independently of spatial location. They also activate partially in response to partial similarity. As a result, the model is order-sensitive without being order-rigid. If the network stabilizes on a pattern which is not a solution, the model identifies a symbol ordering consistent with the current state and uses this symbol ordering to reset the state for another episode of stabilization. This model correctly predicts the positive correlations between high bigram frequency in the target and solution time, and between low bigram frequency in the stimulus and solution time. However, because the constraint satisfaction process is not constrained by lexical information, the model does not seem well-suited to modelling normal reading. The model, NGRAMSWELL, was proposed by this researcher [11].

Authors in [12] carried out experiment to show that type-based(bi-gram of a particular word) bi-gram frequency is a better predictor of the difficulty of anagram solution than is token-based(all occurrences of bigram in a corpus) frequency.

A series of orthographic measures for psycholinguistic research was presented in [13]. Orthographic measure factors are word length, word-form frequency, lemma frequency, neighbourhood density and frequency, transposition neighbours, addition and deletion neighbours are essential.

The research conducted in [14] showed that a new bi-gram frequency measure called top rank, is an important predictor of anagram difficulty, previous researchers opined that the type count are better than the token measures at predicting anagram difficulty. Previous researches suggested many variables that influence the difficulty of anagram i.e. features between anagrams. Methods like Brute force [17], Sorting [8], Bubble sort [11], of solution word, word imagery, correctness of word familiarity, age of acquisition, meaningfulness of vowels, staring letter, some bigram frequency measures. Further research revealed that the most important measure for difficulty is bigram frequency.

Anagram tasks are frequently used in behavioural research to investigate a wide array of cognitive phenomena. Most prominently, they are used to study the cognitive stages involved in problem solving, specifically insight [8].

Researchers in [1] affirmed that anagram solution tasks have been frequently used to assess word recognition processes. Also anagram solution ability is closely related to reading. Reading is an innate skill, it is one of the most crucial cognitive skill that support school based learning. This is because it is a lengthy process that requires mastering a large set of strategies.

Anagram solving task are considered to tap high level cognitive abilities [15]. The anagram paradigm had been used in cognitive research to assess word recognition process.[3]. In anagram solving, bigram frequency is an important feature, anagrams derived from target words with high bigram frequency are easier and vise versa [16].

Some cognitive tasks like reading, spelling, making lexical decision, solving anagram task e.t.c. requires knowledge of orthographic structure of English. Examples of orthographic structure are syllabification, orthographic neighbourhood size, and bigram frequency [12].

The authors in [17] discovered that computer based anagram generators do not provide controls necessary for psychological research. The research used brute force algorithm to automatically process words in a user defined source vocabulary and output, for each word all possible anagrams that exist are defined by the same source vocabulary.

Experiment was conducted to find out cognitive flexibility, this was performed using anagram word puzzles. It was observed that contrast in performance between WAKE, REM(dreaming) and NREM suggests that cognitive processing differs in the three states. It can be inferred from this research that anagram is a true measure of cognitive abilty [18].

Researches on anagram had explored different methods for detecting orthographic similarity

Neighbourhood frequency i.e. counting and histogram [1, 7, 9].

Table 1. Previous Approaches to Orthographic Analysis of Anagram Task

S/N	Authors	Method	Pro	Cons
1	Menalaos and Chris [1]	Syllabic structure and grapheme frequency of target words on anagram solving task	It unveiled how: -Syllabic structure affects anagram solving time -Syllabic complexity has effect on reading fluency measure	The correlation between the recognition of anagram containing infrequent grapheme and those with frequent phoneme-to-grapheme mapping is not explored
2	Norvick and Shennem [12]	Position-sensitive type-based bi-gram frequency	It predicts the difficulty of anagram solution compared to token based numbers	It is restricted to five letters word
3	Sergio et al. [4]	Applied rasch scaling to syllabic structure	To establish individual's ability to solve anagram and the relative difficulty of anagram	Syllable number is restricted to five letters.
4	Hua-Zhan Yin et al. [6]	Syllabic solution of anagram	It depicts unconscious and conscious error detection in anagram solution task	Orthography is restricted to Chinese language.
5	Couriear and Lequexx [7]	Orthographic neighbourhood size	Established relationship between lexical units, prime and target stimuli	Research was conducted using analysis tool. There was no standard software for implementation
6	Mary et. al. [9]	Syllabic structure and orthographic neighbourhood frequency	-Syllabic structure was used to compare the effect of memory on copying words -Neighbourhood frequency was used to compare memory for anagram	Research was conducted using analysis tool. There was no standard software for implementation
7	Sandra et. al. [10]	Exploration of syllabic structure	It is used to evaluate the cognitive ability of patients with primary progressive aphasia	Anagram test was conducted orally
8	Robert et. al. [8]	Syllabic structure using sorting anagram detection technique	It enables direct control of psycholinguistic features that may influence the cognitive process in anagram solution	-Bigram frequency calculation was not directly incorporated in the software -Orthographic neighbourhood frequency is not explored

9	Henin et. al. [11]	The orthographic structure used is N-GRAM WELL, with bubble sort as a measure of anagram detection	It serves as better predictor of anagram detection. This is done by using bubble sort to find the distance transforming a supplied anagram answer to target word. -It is position free	Bubble sort makes execution using N-Gram slower
11	Matthew [18]	Syllabic structure rearrangement	Anagram was used to predict cognitive flexibility	Anagram task was restricted to five letters.
12	Thimothy and Axel [17]	Anagram permutation algorithm(Brute force) using syllabic structure rearrangement	Words can be processed automatically in user defined source vocabulary and output.	Brute force has a poor running time

Table 1 shows that researchers used one or more orthographic structure for anagram analysis. The following are draw backs of previous researches.

- Restriction of anagram letters to five [4, 12]
- Use of analysis tool, no standard software was developed [7, 9].
- Oral conduction of anagram test, no standard software was developed [10].
- The existing cognitive software does not have Bigram orthographic structure incorporated into it. It uses sorting detection technique i.e. no syllabic structure relationship detection [8]
- Bi-gram frequency with bubble sort anagram detection without consideration of position of characters [11]
- Poor anagram detection processing time [11, 17].

3.1 Existing Anagram Orthographic Similarity Measures

The presence of anagram within two strings can be detected by using any of the following methods:

3.1.1 Brute force: This has to do with listing all permutations of the first string, and check if the second string is equal to any of the permutations of the first. This gives a very poor complexity n!. The algorithm for brute force is given as follows [19]:

```
private boolean areAnagrams(String first, String second) {

return areAnagramsRec("", first, second); }

private boolean areAnagramsRec(String soFar, String remaining, String target) {

 if (remaining.length() == 0)

{ return soFar.equals(target); }

for (int i = 0; i < remaining.length(); i++)

 { String whatsLeft = remaining.substring(0, i)+remaining.substring(i+1);

if (areAnagramsRec(soFar + remaining.charAt(i), whatsLeft, target))

return true; } return false; }
```

3.1.2 Sorting: Two strings are anagrams of each other if they are equal, when their letters are sorted. Thus, the presence of anagram can be detected by sorting the characters in each string, and test if the characters in both strings are equal. The method for anagram detection for sorting (ADTS) is given as follows [19]:

```
private boolean areAnagrams(String first, String second)
 { char[] one=Arrays.sort(first.toCharArray());
char[] two = Arrays.sort(second.toCharArray());
return Arrays.equals(one, two); }
```

$$ADTS(x,y)=1-\left[\left|S_x\right|,\left|S_y\right|\right]-\left[\left|S\right|_i\bigcap\left|S\right|_j\right]\text{...........}1$$

where S_i and S_j are characters of sorted list of strings X and Y.

$|S_x|$ and $|S_y|$ represent length of sorted list X and Y

$\left[\left|S\right|_i\cap\left|S\right|_j\right]$ finds out the presence of a character in both strings, and add 1 to sum for each presence. The sum of the intersection must be equal to the length of X as well as Y for anagram to hold.

3.1.3 Counting: It has to do with counting how many times each character occur in each string, and confirm that each string has the same number of character as the other. The method for anagram detection for counting (ADTC) is given as follows [19]:

```
  private boolean areAnagrams(String first, String second) {
 for (char ch = 'a'; ch <= 'z'; ch++) {
if (numCopiesOf(ch, first) != numCopiesOf(ch, second))
 { return false; } return true; }
private int numCopiesOf(char ch, String str) {
 int result = 0;
for (int i = 0; i < str.length(); i++) {
if (str.charAt(i) == ch) result++; }
 return result; }
```

The algorithm above is interpreted into the following mathematical equation for easy evaluation.

$$ADTC(x,y)=1-\left|\left|C\right|_i-\left|C\right|_j\right|_{i=0,\,j=0}^{n,m}\text{.................}2$$

C – represents the number of count of each character i.e. frequency

If the strings are anagram the result will be one, because the number of counts of all characters will be zero.

3.1.4 Histogram: This works by building frequency histogram of characters in each string, and check whether both strings are the same. This method is not explored because it's execution time is similar to counting method.

The methods listed above i.e. brute force, sorting, neighbourhood frequency: counting and histogramming were explained above. Their complexities are O(n!), O(nlogn), O(n^2) ,O(n). [19] Thus, brute force runs with the poorest running time, followed by counting, sorting and then histogramming. The research is thus considering sorting and counting, these two are preferred because histogram is faster but it has trade off for space, while brute force is too poor in terms of running time.

The techniques analyse anagram in words by considering the occurrence of characters without taking cognisance of position of characters. For example, to analyse the strings "ALLERGY" and "ALLERGY" anagrammatically, the existing methods test for the occurrence of all the characters in the first and in the second word without considering their positions. The two methods adopted in this work are ADTS and ADTC. Thus, these mesures analyses two equal strings as anagram.

3. RESULTS AND DISCUSSION

3.1 Implementation and Results

The methods were implemented using java programing language, the back end was implemented using SQLite studio. Eigthy seven unique words were autographically examined. Correct anagrams were extracted from each word and populated in the anagram dictionary which serves as the dataset. Experiment was conducted many times with different dataset. Experiments were carried out on the dataset using the following procedures:

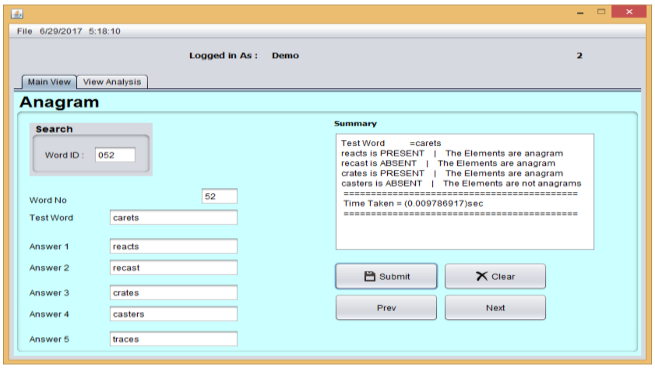
Figure 1 is the interface for analysing a given string and supplied string by user for anagram relationship. A word is selected from the database, and the user is to supply one to five anagrams depending on the number of anagrams that can be generated from a word. The submit button triggers anagram test for cognitive ability. Anagram test is performed using sorting and counting methods, these methods were used to compare the orthographic structure of strings using syllabic frequency check. The methods considers the occurrence of all characters in sting1 and string2, without considering the position of characters which is key in anagram detection. ADTS and ADTC takes care of anagram of the same length, they both returns value 1 in this case, indicating that anagram is detected. If the text and pattern are equal they evaluate to 1, because they check for character occurrence. In ADSM, it's value is either 0 or 1, thus, if text and pattern has unequal length and equal strings, it evaluates to zero, and strings with same character composition but different position evaluates to 1. ADTS and ADTC does not have any function incorporated into it, for detection of wrong anagram lexemes supplied by students i.e. orthographic analysis.

Table 2 depicts the result of anagram test of strings (sample words) using different conditions, this is to show the reaction of these methods to the stipulated conditions. The conditions are explained below:

1. Text longer than Pattern: A string of shorter length was supplied as anagram on serial number 2 i.e. pattern. All the methods evaluates it to zero. Meaning that it's not an anagram
2. Text shorter than Pattern: A string of Longer length was supplied as anagram on serial number 4. All the methods evaluates it to zero. Meaning that it's not an anagram
3. Text equivalent to Pattern: Exactly the same string was supplied as anagram on serial number 6. ADTS and ADTC evaluate to 1 because it sees them as anagram, these methods does not consider position of strings, rather it considers occurrence of strings.
4. Absence of Pattern in the dictionary(user defined vocabulary): A wrong string was supplied as anagram on serial number 8, ADTS and ADTC considers it as anagram because it doesn't do correct word verification before returning output.
5. Absence of character entailment between Text and Pattern: A wrong word was
6. supplied as anagram on serial number 10. The methods returned 0.

Table 2 shows the anagram status and processing time of each text and pattern. The processing time of the dataset for each method was averaged, this indicates that ADTS is the faster, followed by ADTC, just as revealed by the method's complexity.

Experiments on the measures revealed that, tested measures only test for anagram through character entailment verification. Moreover, in anagram, position of characters is very important, these measures doesn't do position verification, which might amount to analysing same words presented as given and target words as anagram which is the case in Table 2, No. 6. Also, apart from conducting orthographic anagram similarity measure, these measures can not test for syllabic relationship between anagram strings, which actually reveal the distance between two anagrams, and thus unveil their complexity i.e. simple, average or hard anagram.



File 6/29/2017 5:18:10

Word ID : 052

Word No 52

Test Word carets

Answer 1 reacts

Answer 2 recast

Answer 3 crates

Answer 4 casters

Answer 5 traces

```
Test Word        =carets
reacts is PRESENT   |   The Elements are anagram
recast is ABSENT   |   The Elements are anagram
crates is PRESENT   |   The Elements are anagram
casters is ABSENT   |   The Elements are not anagrams
=============================================
Time Taken = (0.009786917)sec
=============================================
```

Figure 1: Main View of Orthographic Anagram Analysis Interface

Table 2: Anagram Methods Running Time Analysis Table for Sample Words

S/NO.	TEST WORD/ USER'S ANSWER	ADTS Anagram value; Processing Time (*10^-^{5})secs	ADTC Anagram value; Processing Time (*10^-^{5})secs
1	Auctioned/Cautioned	1; 2.3	1; 65.0
2	Auctioned/educatio	0; 2.3	0; 65.0
3	Allergy/Regally	1; 1.9	1; 07.0
4	Allergy/Galllery	0; 1.9	0; 07.0
5	Antler/Rental	1; 2.3	1; 7.5
6	Antler/Antler	1; 2.3	1; 7.5
7	Ales/Leas	1; 1.5	1; 3.3
8	Ales/Laes	1; 1.5	1; 3.3
9	Assert/Asters	1; 1.8	1; 4.4
10	Asserts/Strong	0; 1.8	0; 4.4
	Mean	1.78	17.44

4. CONCLUSION

Previous approaches to orthographic similarity of anagrams were based on Brute force, Sorting, Orthographic neighborhood frequency. The experimented measures were picked based on their running time complexities and memory space. Data sets were tested on the measures for various conditions to reveal the behaviour of the measures under different circumstances. User defined vocabularies and orthographic parameters was used for orthographic verification. Experiment revealed that the measures has the capacity to test for orthographic similarity of anagram through character entailment verification only. The drawback thus, lies in lack of character position verification and syllabic relationship test which are very vital while testing user's working memory capacity.

ACKNOWLEDGEMENTS

We want to use this medium to acknowledge the efforts of the laboratory technicians of Computer Science Department, Lagos State University for their technical support in the course of this research, i.e. Mr Mayowa and Mr Bamiro. We also want to commend the effort of Mr Ekundayo a Software analyst for his technical support during the program implementation phase of this research. This study is self-sponsored.

AUTHORS' CONTRIBUTIONS

Raji-Lawal Hanat Y. designed the study, performed the statistical analysis, wrote the protocol, and wrote the first draft of the manuscript. Akinwale Adio T. Folorunsho O., and Mustapha Amidu O. conducted literature searches for the study. All authors read and approved the final manuscript."

REFERENCES

1. Menelaos E.S and Chris T.P, . *Linguistic effect on anagram solution: The case of transparent language.* world class journal of education, 2013. **3**(4): p. 41-51.
2. Fink, T.E. and R.W. Weisberg, *The use of phonemic information to solve anagrams.* Memory & Cognition, 1981. **9**(4): p. 402-410.
3. Cornelissen, P., et al., *Coherent motion detection and letter position encoding.* Vision Research, 1998. **38**(14): p. 2181-2191.
4. Sergio P., A.O., Juan E.T., Pedro P., *Randomized anagram revisited.* Cosec Laboratory, Elsevier, 2014.
5. Adams, J.W., et al., *The role of syllables in anagram solution: A Rasch analysis.* The Journal of general psychology, 2011. **138**(2): p. 94-109.
6. Hua-Zhan Yin, J.Y., Wei Li, Jiang Qui, Ling Yu Chen *Neural bases of unconscious error detection in a Chinese anagram solution task: Evidence from ERP study.* Journal Pone 0154379, 2016.
7. Lequex, C.P.a., *anagram effect in visual word recognition*. 2004, https://archives-onverts.fr/hat0042984.
8. Robert D.V, Y.K.G., Debra A.T *Anagram software for cognitive research that enables specification of Pscholinguistic variables.* Behaviour Research Methods, Instruments and Computers, http://dx.doi.org/10.3758/BF03195587, 2006. **38**(2).
9. Mary A F, H.J.F., Alice W., Leslie R. , *Anagram solving: Does effort have an effect.* memory and cognition, 1989. **17**(6): p. 755-758.
10. Sandra W., M.-M.M., Christina W, Alfred R, Emily J.R, Cynthia K.T. , *The Northerwestern Test: Measuring sentence production in primary progressive Aphasia.* American Journal of Alzhheinder's disease and other Dementias, 2009.
11. Henin, J., et al. *Extraordinary natural ability: Anagram solution as an extension of normal reading ability*. in *Proceedings of the 31st Annual Meeting of the Cognitive Science Society*. 2009.
12. S.J., N.L.R.a.S., *Type based bigram frequencies for five letter words.* Behaviour Research Methods, Instruments and Computers, http://dx.doi.org/10.3758/BF03195587, 2004. **36**(3): p. 397-401.
13. Maria Ktori, W.J.B., Van Heuven, Nicola J Ptchford, *Greek Lex: A lexical database of modern greek.* Behaviour Research Methods, 2008. **40**(3): p. 773-783.
14. David Knight, S.J.M., *Type and token bigram frequencies for two trough nine letter words and the prediction of anagram difficulty.* Behaviour Research Journal, 2011. **43**: p. 491-498.
15. Mozer, G.a., *The interplay of symbolic and sub-symbolic processes in anagram problem solving.* Advances in neural information processing system 2001. **13**: p. 17-23.
16. Johnson, G., *Effect of solution word attributes on anagram difficulty.* A regression analysis quarterly journal of experimental psychology, 1978. **30**: p. 57-70.
17. Monteiro, T.R.J.a.A., *Generating anagrams from multiple core strings employing user defined vocabularies and orthographic parameters.* behaviour Research Methods, Instruments and Computers, 2003. **35**(1): p. 129-135.
18. Matthew P. Walker, C.L., J. Allan Hobson, Robert Stick Gold, *Cognitive flexibility across sleep-wake cycle: REM-Sleep enhancement of anagram solving.* Elsevier Journal of cognitive brain research, 2002. **14**: p. 317-324.
19. University, S., *Anagram detection* web.stanford.edu/class/cs9/Sample-Probs/Anagram.pdf