JOURNAL OF APPLIED
COMPUTER SCIENCE
Vol. 23 No. 1 (2015), pp. 7-28

Efficient Similarity Measures for Texts Matching

Adio Akinwale[1], **Adam Niewiadomski**[2]

[1]*Federal University of Agriculture*
Department of Computer Science
P. M. B. 2240, Abeokuta, Nigeria
aatakinwale@yahoo.com

[2]*Lodz University of Technology*
Institute of Information Technology
Wólczańska 215, 90-924 Łódż, Poland
adam.niewiadomski@p.lodz.pl

Abstract. *Calculation of similarity measures of exact matching texts is a critical task in the area of pattern matching that needs a great attention. There are many existing similarity measures in literature but the best methods do not exist for closeness measurement of two strings. The objective of this paper is to explore the grammatical properties and features of generalized n-gram matching technique of similarity measures to find exact text in electronic computer applications. Three new similarity measures have been proposed to improve the performance of generalized n-gram method. The new methods assigned high values of similarity measures and performance to price with low values of running time. The experiment with the new methods demonstrated that they are universal and very useful in words that could be derived from the word list as a group and retrieve relevant medical terms from database . One of the methods achieved best correlation of values for the evaluation of subjective examination.*

Keywords: *similarity measures, fuzzy relations, n-gram, word list, set theory, subjective examination.*

1. Introduction

Measuring the similarity among sets of texts is very essential in various tasks such as information retrieval, document categorization and plagiarism detection. Selection of relevant document through similarity measure is fundamental on these applications but important task. In order to overcome these tasks, researchers have proposed several methods for measuring similarity that would locate exact fragments of text from the pool of text. While many similarity measures have been proposed and individually evaluated, they have not been tailored to each other in a large real-world environment. A growing number of tasks especially those related to web search technologies rely heavily on accurately computing the similarity between two segments of text. Finding a suitable similarity measure is often the most critical part of electronic web search technologies.

Many program languages provide in-built functions that work perfectly. In case, there is a need to know if one string is very close to another but not equal, programming languages do not have in-built functions for closeness measurement. There is no way to verify in programming languages if two strings are seventy two percent very close to each other. Moreso, when two strings are not hundred percent equal due to mismatch errors, things get a little more complicated. It is a general belief to rely on fuzzy logic to find the correct percentage of matches or mismatches. The selected matches or mismatches should be verified and approved manually. Another problem is when the user need to find records or information that satisfy a similarity predicate while exact matching is not suffiicient. These queries are very important for web application where errors abbreviation and inconstencies are very common. Similarly in electronic test, it is also possible that a student would not able to obtain hundred percent answer and exact score must be given based on the expert judgement. This normally happens in subjective examination where students make spelling errors in their answers and a fair judgment is required. We may wish to find all the facebook users who have similar friends in a web site. For instance, a mobile or normadic medical doctor may wish to prescribe an urgent drug but fail to know the exact spelling. To meet these types of needs, a good similarity measure is required. This absence of close measurement of two strings is noteworthy. There are a large number of similarity measures proposed in literature but the best similarity measures do not exist for closeness measurement of a paricular domain.

2. Literature reviews

Essentially, the n-grams model is a probabilistic model originally devised by the Russian mathematician, Andrey Markov in the early 20th century and later extensively experimented by Shannon and Chomsky for predicting the next item in a sequence of items [1]. The first use of n-gram dates to world war second when it was used by cryptographers. Fletcher Pratt stated that with the backing of bigram and trigram tables, any cryptographer can dismember a simple substitution cipher [2].

N-grams have been successfully used for a long time in a wide variety of problems and domains such as text compression [3], spelling error detection and correction [4], optical character recognition, information retrieval [5], automatic text categorization, music representation, speech and handwriting recognition [6]. Other useful domains include computational immunology, analysis of whole-genome protein sequences [7], language identification, authorship attribution, phylogenetic tree reconstruction, data integration, filtering and cleaning, prediction of English Language [8], phonetic matching algorithms, and text retrieval [9].
A typical example, acceleration of general string searching has been accomplished using n-gram signatures by Harrison in 1971 [10]. Assale et. al. addressed an n-gram based signature method to detect computer viruses [11]. N-gram methods have proven to be useful in a variety of tasks ranging from comparison of two texts to the quantification of degrees of homology in genetic sequence. The method is widely used for solving problems in different areas such as operation research, computer science, biology, etc. Arsmah had used n-gram to grade mathematic texts [12]. His research showed that n-gram method proved to be suitable when applied to the four linear algebraic equations. Ukkonen used the sum absolute different between corresponding numbers of n-gram occurrence in each string for approximate string matching. Alberto Barman-Cedeno and Paolo Rosso used n-gram to determine if a given text is plagiarized from the pool of METER corpus [13]. Prahlad [14] stated that n-gram was used for intrusion detection whereby the system relies on substring match of network traffic or host activities with normal patterns or attack patterns. Niewiadomski used generalized n-gram matching for automatic evaluating text examination and employed the method also to evaluate electronic language test using German Language as a case study [15]. Chask found charater n-grams to work well for attribution in a forensic context [16]. All the results from the application of n-gram prove positive and there is a need to do more research on it for further improvement.

3. Similarity as a relation

Definition 1 *A similarity relation on a set U is a fuzzy binary relation* $R : U \times U \longrightarrow [0,1]$ *holding the following properties:*

$$Reflexive \qquad R(x,x) = 1 \quad for\,any\,x \in U \tag{1}$$

$$Symmetric \qquad R(x,y) = R(y,x) \quad for\,any\,x,y \in U \tag{2}$$

$$Transitive \qquad R(x,z) \geq R(x,y) \vartriangle R(y,z) \quad for\,any\,x,y,z \in U \tag{3}$$

where the operator $\vartriangle$ is an arbitrary t-norm. A t-norm $\vartriangle : [0,1]\times[0,1] \longrightarrow [0,1]$ is a binary operator which is commutative, associative, monotone in both arguments and $1 \vartriangle x = x$ hence it subsumes the classical two-valued conjunction operator.

We consider a relation of similarity x_1 and x_2 which is written as $x_1 \sim x_2$. These similarity relations are subject to reflexive and symmetry and may not be necessarily be transitive. In this case, relation R on X is called the relation of neighbourhood if R is reflexive on X and R is symmetry on X. Neighbourhood relationship is also referred as follows: non-sup-min transitive similarity relation, tolerance relation, proximity relation, partial preorder relation, resemblance relation, approximate equality relation, etc.

3.1. Set similarity

Definition 2 *Definition: Let A, B be arbitrary sets on X.*

Function $\mu : X \longrightarrow R^+ \cup 0$ is a measure of sets if and only if

$$\mu(\phi) = 0 \tag{4}$$

$$\mu(A \smile B) \leq \mu(A) + \mu(B) \tag{5}$$

Equations 4 and 5 can narrow down to $\mu(A \smile B) = \mu(A) + \mu(B) - \mu(A \frown B)$
Other properties of similarity measure sets are:

$$A = B \longrightarrow \mu(A) = \mu(B) \tag{6}$$

$$A \subseteq B \longrightarrow \mu(A) \leq \mu(B) \tag{7}$$

The implication of the reverse is not necessarily to be true [15]. The focus would be on the intuition that the degree of similarity should take into account

both the amount of overlap between the given sets and the amount of symmetric difference. The following formulae presented as models of perceptual similarity in crisp setting clearly reflect the ideas behind the construction of similarity measures in set theoretic contexts. These measures were proposed by Tversky [17] [18]. For any two sets A and B
1. $S(A,B) = \theta f(A \cap B) - \alpha f(A-B) - \beta f(B-A)$. where f(.) is usually the cardinality of the set.
2. $S(A,B) = \frac{f(A \cap B)}{f(A \cap B) + \alpha f(A-B) + \beta f(B-A)}$, where the value is normailized to the range [0, 1].

If symbols ∪ and ∩ are modelled by max and min t-norms and ∇ is defined as: $A \nabla B(x) = max[min(A(x), 1 - B(x)), min(B(x), 1 - A(x))]$ then the following are the set-theoretic similarity measures for fuzzy sets presented in [19] [20] :
1. The analogous of Restle's model : $S(A,B) = 1 - \mid A \nabla B \mid$
2. The analogous of Gregson's model : $S(A,B) = \frac{|A \cap B|}{|A \cup B|}$
3. The analogous of Enta's model : $S(A,B) = sup_{x \in X} A \cap B(x)$
The most important thing to be noticed about these measures is that they are not necessarily t-transitive in nature unlike distance based similarity measures.

If d is the distance measure between two fuzzy sets A and B on a universe X, the following similarity measures are presented in respectively:
1. The distance based assessment proposed by Koczy: $(A,B) = \frac{1}{1+d(A,B)}$
2. The distance based assessement proposed by Williams and Steele : $S(A,B) = e^{\alpha d(A,B)}$ where α is the steepness measure.
3. Family of distance based similarity measures presented by Sanitni : $S(A,B) = 1 - d_r(A,B)$, $r = 1, 2, \cdots, \infty$ [21].

4. N-gram method

Definition 3 *Let* $A = (a_1, a_2, a_3, \ldots, a_n)$ *be a sequence, where* $a_i \in \sum(i = 1, 2, 3, \ldots, k)$ *then* $(a_{j+1}, a_{j+2}, \ldots, a_{j+n}) \in \sum^n$ *is called a n-gram of the sequence, where* $0 \leq j \leq k - n$ *: the set of all the n- grams of sequence is called the n-gram set of sequence, that is* $G(A,n) = (a_{j+1}, a_{j+2}, \ldots, a_{j+n}) \mid 0 \leq j \leq k - n$ *is the n-gram set of sequence where* $n \in Z^+$ *is the length of the n-gram. It is noted that* $G(A,n) = \phi$ *if* $k < n$.

Similarity of two strings s_1 and s_2 can be determined via the n-gram method as follows:

$$sim(s_1, s_2) = \frac{1}{N - n + 1} \sum_{i=1}^{N-n+1} h(i) \tag{8}$$

where h(i) = 1 if n-element subsequence beginning from position i in s_1 appears in s_2 h(i) = 0 otherwise;
N-n+1 = number of n-element subsequence in s_1.

4.1. Generalized n-gram matching for string matching

Generalized n-gram matching was introduced by Niewiadomski. The algorithm matches an answer string to a template string as follows [22] :

$$sim(s_1, s_2) \; = \; f(n_1, n_2) \sum_{i=n_1}^{n_2} \sum_{j=1}^{N-n+1} h(i, j) \tag{9}$$

where $f(n_1, n_2) \; = \; \frac{2}{(N-n_1+1)(N-n_2+2)-(N-n_2+1)(N-n_1)}$ denotes the number of possible substrings not shorter than n_1 and not longer than n_2 in $s_1, h(i, j) \; = \; 1$ iff an i-element-long substring of the string s_1 starting from j-th position in s_1 appears (at least) once in s_2 (otherwise $h(i, j) \; = \; 0$). If all substrings from one argument of comparison are found in the other, the final similarity degree is evaluated as 1 which is interpreted as the identity of s_1 and s_2 [22].
$N(s_1), N(s_2)$ = length of string s_1 and s_2 ,
N = $max(N(s_1), N(s_2))$

4.2. Bigram method

Generalized n-gram matching is normally used to derive bigram where n is equal to 2, hence the function is as follows:

$$sim(s_1, s_2) \; = \; \frac{1}{N - n + 1} \sum_{i=0}^{N-n+1} h(i) \; = \; \frac{1}{N - 2 + 1} \sum_{i=0}^{N-2+1} h(i) \; = \; \frac{1}{N - 1} \sum_{i=0}^{N-1} h(i) \tag{10}$$

4.3. Trigram method

Two string s_1 and s_2 are determined via the n-gram method as trigram when n is equal to 3. The function is as follows:

$$sim(s_1, s_2) = \frac{1}{N-n+1}\sum_{i=0}^{N-n+1} h(i) = \frac{1}{N-3+1}\sum_{i=0}^{N-3+1} h(i) = \frac{1}{N-2}\sum_{i=0}^{N-2} h(i) \tag{11}$$

5. New methods

5.1. Oddgram method

Oddgram was inspired by the generalized n-gram matching which takes n(n-1)/2 substrings for processing before measuring the performance. The oddgram would take half substrings of generalized n-gram matching for processing the performance which would still reduce the running time. For the method, the matched strings are denoted as s_1, s_2 and $max(N(s_1), N(s_2)) = N$ which is the maximum length between string s_1 and s_2. If N is odd then $N = \lceil \frac{N}{2} \rceil$

$$sim(s_1, s_2) = \frac{1}{N^2}\sum_{i=N}^{N}\sum_{j=1}^{N-i+1} h(i, j) \qquad else \qquad \frac{1}{N^2+N}\sum_{i=N}^{N}\sum_{j=1}^{N-i+1} h(i, j) \tag{12}$$

5.2. Sumsquare gram method

Likewise oddgram, sumsquare gram was inspired by the generalized n-gram matching in which processing time is quadratic for every n-gram in the query string of line statement. While similarity measures of n-gram are easy to generate and manage, they do require quadratic time and space complexity and therefore ill- suited to both oddgram and sumsquare gram which work in quadratic. Oddgram and sumsquare gram methods are expected to write their results into similarity measure (s) between a pair of submissions (pattern matching and text matching). Given pattern matching and text matching i and j, s_{ij} will be near to 1 if both patterns are considered identical and near to 0 if they are very dissimilar. That is, oddgram and sumsquare grams are normalized to fall within the interval [0, 1]. Similarly, similarity measure of oddgram and sumsquare gram are

expected to be symmetric, that is the equality $s_{ij} = s_{ji}$ is expected to hold for every i, j. For the sumsquare gram, the matched strings are denoted as s_1, s_2 and $max(N(s_1), N(s_2)) = N$ which is the maximum length between string s_1 and s_2.
$N = \lfloor \sqrt{N} \rfloor$
$M = times - to - jump = N - 1$
$P = first - jump = N^2 - (N - 1)^2$

$$sim_{sq}(s_1, s_2) = \frac{6}{N(N + 1)(2N + 1)} \sum_{i=1}^{P} \sum_{j=1}^{M} h(i, j) \tag{13}$$

5.3. Set-based trigram method

Set-based trigram was inspired by the theory of set similarity measure in section 3.1. The method measures the similarity between two sets of entities in terms of the number of common trigram. It inceases the weight of string sharing of pattern and text matching by three times. Set-based trigram is asymmetric because it does not consider (false, false) to be a matched patterns. The method is described as follows:

$$set - based\ \ trigram\ : T(X, Y) = \frac{3(trigram(X \frown Y))}{trigram(X) + trigram(Y) + trigram(X \frown Y)} \tag{14}$$

6. Adaptation of Dice method

Experiment in theory, manual and application with the same data set was conducted using existing methods such as Jaccard, Dice, Overlap and Cosine similarity measures. It was discovered by examples that values of similarity measures of overlap, cosine and Jaccard are more or less the same in some cases. For example, cosine is a montonically decreasing function for interval $[0^o, 180^0]$, so also Jaccard and overlap and their inclusion in our experiments would be redundant.

In evaluating one term against another term, Dice similarity is chosen because it is popular and widely used in analogous text of retrieval systems. This measure takes into account the length of terms. The coefficient value varies between zero and one. If two terms have no characters in common then the coefficient value is zero. On the other hand, if they are identical, the coefficient value will be one [21]. For two string X and Y, the Dice coefficient is measured as

$$d(X,Y) = \frac{2(n - gram(X \frown Y))}{(n - gram(X)) + (n - gram(Y))} \tag{15}$$

7. Word list similarity

A word list is the words that are derived from the same root as a given word. The similarity methods would rank the words in the word list in either descending or ascending order of their similarity to the given word. For example, given the word *eloquently*, the similarity measures are to retrieve the other related words such as *ineloquently, ineloquent, eloquent, eloquence,* etc. Other similar words of the word list are: *president, presidency, presidential, etc, program, programmer, programming, etc ,* Samples of these words of word list are calculated manually with known and proposed methods as shown in example 1 and 2 .

Example 1:

Let s_1 = ELOQUENTLY, s_2 = INELOQUENT. N(s_1) =10 and N(s_2) =10, $max\{N(s_1), N(s_2)\} = 10$

s_2 occurs in the substring of s_1 as follows:

9 1-element E, L, O, Q, U, E, N, T = 8
7 2-element EL, LO, OQ, QU, UE, EN, NT, = 7
6 3-element ELO, LOQ, OQU, QUE, UEN, ENT = 6
5 4-element ELOQ, LOQU, OQUE, QUEN, UENT = 5
4 5 -element ELOQU, LOQUE, OQUEN, QUENT = 4
3 6-element ELOQUE, LOQUEN, OQUENT = 3
2 7-element ELOQUEN, LOQUENT = 2
1 8 -element ELOQUENT = 1

1. Generalized n-gram matching

$$sim(s_1, s_2) = \frac{2}{N^2+N} \sum_{i=1}^{N} \sum_{j=1}^{N-i+1} h(i, j) = \frac{2}{10^2+10} \times \frac{8+7+6+5+4+3+2+1}{1} = \frac{2*36}{110} =$$

0.65

2. Dice's Coefficient = $d(X, Y) = \frac{2(n-gram(X \frown Y))}{n-gram(X)+(n-gram(Y))} = \frac{2(7)}{9+9} = \frac{14}{18} = 0.77$

3. Bigram = $sim(s_1, s_2) = \frac{1}{N-n+1} \sum_{i=0}^{N-n+1} h(i) = \frac{1}{10-1} \times \frac{7}{1} = \frac{7}{9} = 0.77$

4. Trigram = $sim(s_1, s_2) = \frac{1}{N-n+1} \sum_{i=0}^{N-n+1} h(i) = \frac{1}{10-2} \times 61 = \frac{6}{8} = 0.75$

5. $Set - based\,trigram = sim(s_1, s_2) = \frac{3*(trigram(s_1 \frown s_2)}{trigram(s_1)+trigram(s_2)+trigram(s_1 \frown s_2)} = \frac{3*6}{8+8+6} = \frac{18}{22} = 0.82$

6. Oddgram = If N is odd then $N = \lceil \frac{N}{2} \rceil = N = even = \lceil \frac{10}{2} \rceil = 5$, $sim(s_1, s_2) = \frac{1}{N^2+N} \sum_{i=N}^{N} \sum_{j=1}^{N-i+1} h(i, j) = \frac{1}{5^2+5} \times \frac{7+5+3+1}{1} = \frac{16}{30} = 0.53$

7. sumsquare gram = $N = \lfloor \sqrt{N} \rfloor = 3$, $M = timestojump = N - 1 = 2$, $P = firstjump = N^2 - (N-1)^2 = 3^2 - 2^2 = 5, 2^2 - 1^2 = 3$, $sim_{sq}(s_1, s_2) = \frac{6}{N(N+1)(2N+1)} \sum_{i=1}^{P} \sum_{j=1}^{M} h(i, j) = \frac{6}{3(4)(7)} \times \frac{8+3+0}{1} = \frac{11}{14} = 0.78$

Example 2:

Let s_1 = PROGRAMMER, s_2 = PROGRAMMING. N(s_1) =10 and N(s_2) =11, $max\{N(s_1), N(s_2)\} = 11$

s_2 occurs in the substring of s_1 as follows:

9	1-element	P, R, O, G, R, A, M, M, R = 9
7	2-element	PR, RO, OG, GR, RA, AM, MM = 7
6	3-element	PRO, ROG, OGR GRA, RAM, AMM = 6
5	4-element	PROG, ROGR, OGRA, GRAM, RAMM = 5
4	5 -element	PROGR, ROGRA, OGRAM, GRAMM = 4
3	6-element	PROGRA, ROGRAM, OGRAMM = 3
2	7-element	PROOGRAM, ROGRAMM = 2
1	8 -element	PROGRAMM = 1

1. Generalized n-gram matching

$sim(s_1, s_2) = \frac{2}{N^2+N} \sum_{i=1}^{N} \sum_{j=1}^{N-i+1} h(i, j) = \frac{2}{11^2+11} \times \frac{9+7+6+5+4+3+2+1}{1} = \frac{2*37}{132} = 0.56$

2. Dice's Coefficient = $d(X, Y) = \frac{2(n-gram(X \frown Y))}{n-gram(X)+(n-gram(Y))} = \frac{2(7)}{9+10} = \frac{14}{19} = 0.74$

3. Bigram = $sim(s_1, s_2) = \frac{1}{N-n+1} \sum_{i=0}^{N-n+1} h(i) = \frac{1}{11-1} \times \frac{7}{1} = \frac{7}{10} = 0.70$

4. Trigram = $sim(s_1, s_2) = \frac{1}{N-n+1} \sum_{i=0}^{N-n+1} h(i) = \frac{1}{11-2} \times 61 = \frac{6}{9} = 0.64$

5. $Set - based\, trigram = sim(s_1, s_2) = \frac{3*(trigram(s_1 \frown s_2)}{trigram(s_1)+trigram(s_2)+trigram(s_1 \frown s_2)} = \frac{3*6}{8+9+6} = \frac{18}{23} = 0.78$

6. Oddgram = If is odd then $N = \lceil \frac{N}{2} \rceil = N = odd = \lceil \frac{11}{2} \rceil = 6, \; sim(s_1, s_2) = \frac{1}{N^2} \sum_{i=N}^{N} \sum_{j=1}^{N-i+1} h(i, j) = \frac{1}{6^2} \times \frac{9+6+4+2}{1} = \frac{21}{36} = 0.58$

7. sumsquare gram = $N = \lfloor \sqrt{N} \rfloor = 3, \; M = timestojump = N - 1 = 2, \; P = firstjump = N^2 - (N-1)^2 = 3^2 - 2^2 = 5, \; 2^2 - 1^2 = 3, \; sim_{sq}(s_1, s_2) = \frac{6}{N(N+1)(2N+1)} \sum_{i=1}^{P} \sum_{j=1}^{M} h(i, j) = \frac{6}{3(4)(7)} \times \frac{9+3+0}{1} = \frac{12}{14} = 0.85$

Looking at these two examples, sumsquare gram and set-based trigram have the highest values of similarity measures of 0.85 and 0.78 for the strings of (PROGRAMMER, PROGRAMMING) and 0.78 and 0.82 for the strings of (ELOQUENTLY, INELOQUENT).

8. Experiment

Three types of experiment were conducted for the purpose of testing the performance of the new methods as against the existing ones. The first experiment used words that can be derived fron the group of word list as described in table 1, while the second experiment was conducted on medical database terminologies. The last experiment was based on subjective examination in which fair judgment would be given to students who normally make spelling errors on their answers.

8.1. Experiment one

The new similarity measures of oddgram, sumsquaare gram and set-based trigram together with the existing methods of generalized n-gram, Dice method and bigram were tested for extracting pattern matching from words that can be derived from the word list. Five hundred pairs of words from word list were stored in a file. Sample of the words from word list is illustrated in table 1. This file served as

Table 1. Sample of words from the word list

s/n	word	word list
1	program	programming, programmable, programmer, ...
2	eloquent	eloquently, ineloquent, ineloquently, eloquency, ...
3	fertile	fertilize, fertility, fertilization, fertilizer, ...
4	administer	administration, administrator, administrative,...
5	accounting	accountancy, accountant , accountable, ...
6	question	questionable, questioner, questioning,...
7	responsible	irresponsible, responsibility, responsive,...
8	possible	impossible, possibility, possibly, ...
9	protester	protestant, protestation, protestantism,...
10	inadequate	adequacy, adequately, adequate,...
11	depreciate	appreciation, appreciable, depreciative, ...
12	expedience	expedient, expediently, expediency,...

input into the aforementioned methods. These methods were implemented using JAVA programming language embedded in NetBeans IDE 7.1.2. The experiment was conducted on HP Laptop with an Intel Pentium 2.10 GHz dual core CPU and 1.00 GB memory, running a 32- bit Windows Vista operating system. The average and standard derivation were used. The performance to price (ptp) was measured as (similarity values/running time values). The value of running time has been converted to milliseconds. Due to the number of the words from the word list, the total average of similarity measure, processing time and performance to price was calculated.

8.2. Results

Figure 1, 2 and 3 illustrate the total average similarity, running time and performance to price of the six methods using five hundred pairs of words from the word list. As depicted in figure 1, sumsquare method has the highest value of similarity measure of (0.807), followed by set-based trigram (0.725), dice (0.692), latter by bigram (0.629), oddgram (0.507) and generalized n-gram (0.482). The total average of processing time of sumsquare, set-based trigram, dice and bigram is more or less the same as described in figure 2. Among the six methods, set-based trigram has the highest value of performance to price, followed by dice, sumsquare, bigram, oddgram and generalized n-gram as illustrated in figure 3.

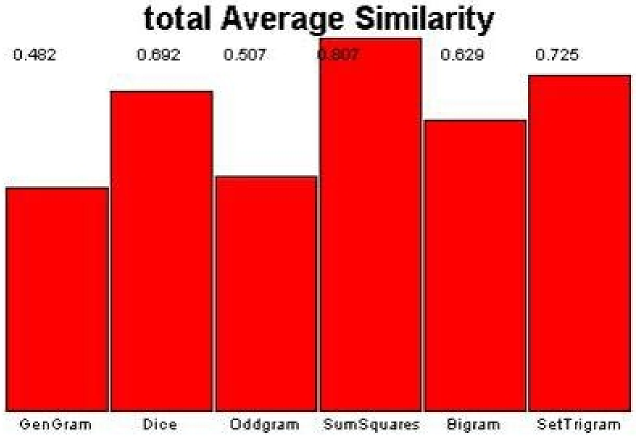


Figure 1. Total average of similarity measures for the methods

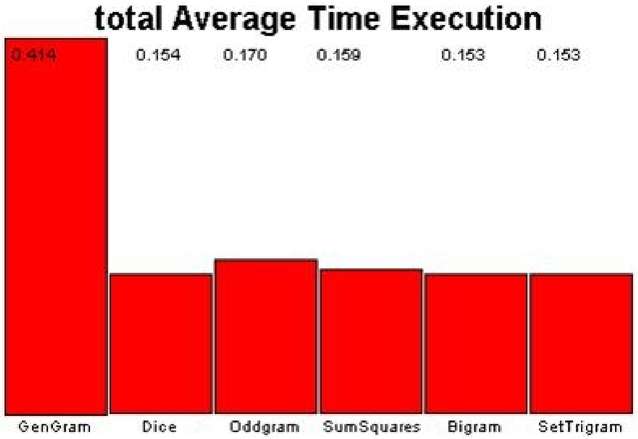


Figure 2. Total average of time execution of the methods

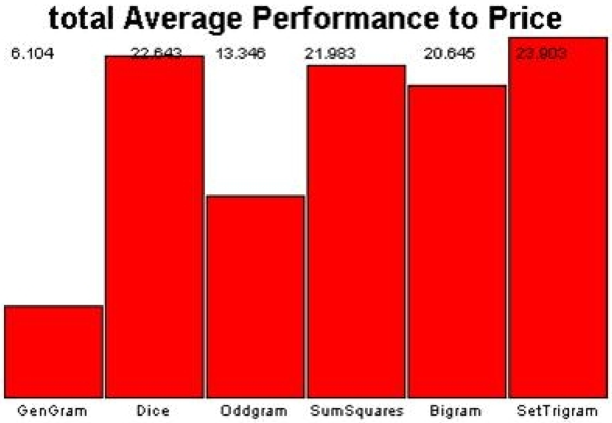


Figure 3. Total average of performanxe to price of the methods

8.3. Experiment two

Similarly, another set of data was collected from the web that contains 600 medical terminologies. These medical terminologies such as belatecan, batracylin, are stored as text database in which they would serve as input to the generalized n-gram, Dice, bigram, oddgram, sumsquare and set-based trigram methods. For example, a medical user would enter medical terminology or part of the terminology as a query matching pattern and the system would select the best appropriate term or near to the term that matches the medical user's requirement. For proper classification of computed pattern matching that similar to text matching entered by the users, k-nearest neighbour (k-NN) method is adopted. The k-NN takes as an input a sparse $| C | \times | C |$ similarity matrix S, where each element s_{ij} represents a semantic similarity of terms $c_i, c_j \in C$. The output is a set of binary relations : $\bar{R} \in C \times C$. It is a standard method which links each term c_i with the k most similar neighbour according to the scores provided in S.

In the experiment, the user is allowed to enter the value for the k. For demonstration purpose in this case, a medical user enter “tazomib” as text matching and matching pattern in the database is “bortezomib”. As shown in figure 4, the value for k is chosen as 0.5 which displays all computed pattern matching above 0.5. Our sumsquare gram method selected appropriate term with similarity value of 0.54 and set-based trigram with similarity value of 0.56 while Dice method value was 0.53. Oddgram selected the medical term with highest similarity value of 0.75 but added some unrelated terms whereas other methods of bigram and generalized n-gram did not select any term. The result is depicted in figure 4.

8.4. Evaluation of the result

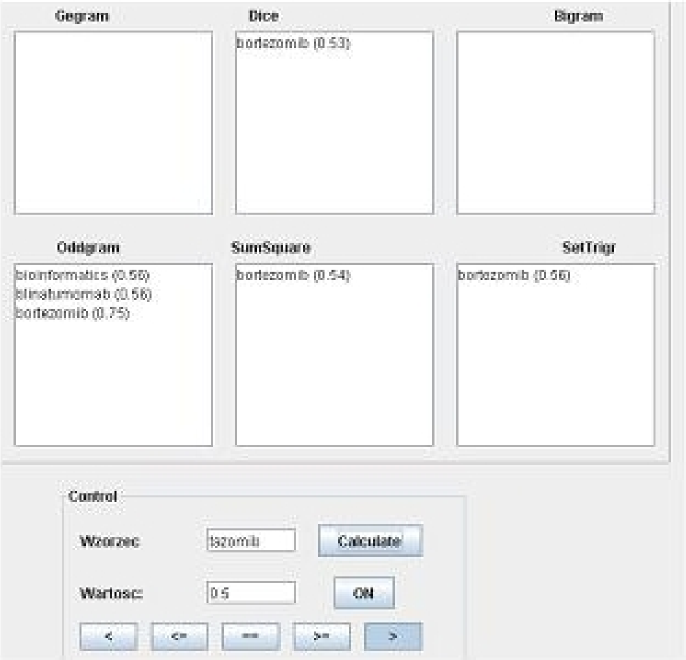
The result generated by the six methods must be evaluated to determine their usefulness in real life of electronic test in the world wide webs. The evaluation employed the function of recall, precision and f-measure to determine to what percentage degree the number of the retrieved and relevant medical terms is proportional closed to the medical users' requests. Table 2 shows the values of recall, precision and f-measure methods of the six methods using medical term database.
Let $c_i, ..., c_n$ denote the retrieved medical terms generated by the six methods.
Let also q_i denote the medical terms retrieved by each method
$c_i(1 <= i < n)$
$R(q_i)$ will contain the set of retrieved medical terms of q_i
Based on this assumption, the methods for precision, recall and f-measure are computed as follows [23]:

$$precision(q_i) = \frac{R(q_i) \cap c_i}{R(q_i)} \tag{16}$$

$$recall(q_i) = \frac{R(q_i) \cap c_i}{c_i} \tag{17}$$

$$f - measure(q_i) = \frac{2 \times precision(q_i) \times recall(q_i)}{precision(q_i) + recall(q_i)} \tag{18}$$

As shown in the table 2, set-based trigram, oddgram and sumsquare methods have the values of 0.923, 0.850 and 0.901 for the recall method which indicate that the methods returned most of the relevant medical terms similar to the users' requests. This is also for the values of set-based trigram, oddgram and sumsquare for precision method of 0.944, 0.861 and 0.910 which indicate that the three methods returned more relevant medical terms than irrelevant medical terms requested by the users. The values of set-based trigram and sumsquare (0.931, 0.905) using f-measure are very close to one (1.00) compared to other methods of generalized n-gram and bigram except Dice method which confirmed the harmonic means of precision and recall values. Thus, set-based trigram and sumsquare methods demonstrated a great improvement in the effective and efficient retrieval of relevant medical terms from the database.



Gegram

Dice
bortezomib (0.53)

Bigram

Oddgram
bioinformatics (0.56)
blinatumomab (0.56)
bortezomib (0.75)

SumSquare
bortezomib (0.54)

SetTrigr
bortezomib (0.56)

Control
Wzorzec: tszomib
Wartosc: 0.5

Figure 4. Extracted values of the six methods

8.5. Experiment three

Data set of post unified tertiary matriculation examination of those seeking admission into Nigerian Universities were used. The data set contains 100 questions (25 questions each) on English Language, Biology, Physics and Chemistry and 12,055 student answers. The questions would allow the student to fill the right answer in the space provided. Samples of the questions are as follows:
(1) ————is an instrument to measure relative humidity (*hygrometer*,higrometer, hygomometer)
(2) In hot weather, the body of a mammal can be cooled throught ———— (*vasoconstriction*,vasocostriction, vasoconstrision)
(3) The disease transmitted to animals by tse-tsefly is ———— (*trypanosonic*, tripanosonic, trypanosomic)
(4) The addition of liming material to the soil is aimed of correcting soil ———— (*alkalinity*, alcalinity, alcalimity)

Table 2. Values of precision, recall and f-measure methods for the six methods

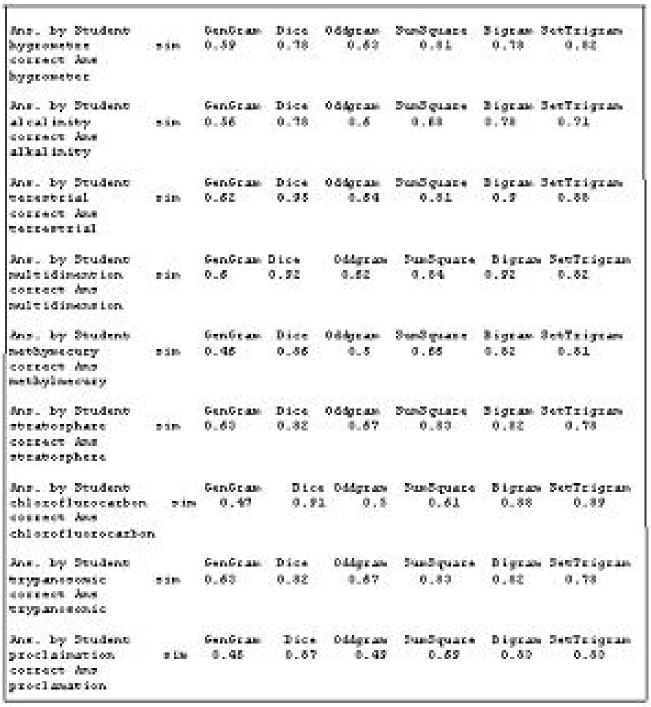
Methods	Gen-n-gram	Dice	bigram	set-trigram	odd-gram	sumsquare
precision	0.408	0.905	0.453	0.944	0.861	0.910
recall	0.505	0.891	0.761	0.923	0.850	0.901
f-measure	0.451	0.897	0.567	0.931	0.855	0.905

(5) ———- is an acid that forms normal salt only (*trioxonitrate*, tryoxonitrate, trioxonstrate)
The italicized words are correct answer while underlined ones are student answers with spelling errors. Randomly sampling of the student answers indicated that some students made a lot of spelling errors such as ommision of letters, subsitution of letters or transposition of letters. For example, the correct answer of question one above is hygrometer while some students wrote hydrometer, higrometer, hygomometer, etc. For the purpose of testing the performance of the methods as illustrated in section 4, the correct answers of the students were filtered out to obtain 7,501 student answers. This number served as input to similarity measures of generalized n-gram, Dice, oddgram, bigram, sumsquare and set-based trigram methods. Figure 5 shows some samples of the similarity values generated by these methods. Looking at figure 5 , the similarity values of oddgram ranges from 0.50 to 0.67, generalized n-gram from 0.46 to 0.63 while from 0.70 to 0.95 are the ranges of Dice, bigram, set-based trigram and sumsquare methods.

8.6. Evaluation of the result

The students' answers that contained spelling errors were given to three experts for scoring. The average for each expert score of 7,501 students' answers was computed. Pearsons correlation coefficient r was used to specify the correlation between automatic scores from similarity measures and average expert grades. The coefficient r is given by:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{(}\sum_{i=1}^{n}(x_i - \bar{x})^2 * \sum_{i=1}^{n}(y_i - \bar{y})^2)} \tag{19}$$



```
Ans. by Student            GenGram  Dice  Oddgram  SumSquare  Bigram SetTrigram
hygrometer           sim   0.59     0.78    0.63     0.81      0.78     0.82
correct Ans
hygrometer

Ans. by Student            GenGram  Dice  Oddgram  SumSquare  Bigram SetTrigram
alcalinity           sim   0.56     0.78    0.6     0.68      0.78     0.71
correct Ans
alkalinity

Ans. by Student            GenGram  Dice  Oddgram  SumSquare  Bigram SetTrigram
terertrial           sim   0.62     0.95    0.84     0.81      0.9      0.88
correct Ans
terrestrial

Ans. by Student            GenGram Dice    Oddgram  SumSquare  Bigram SetTrigram
multidimention       sim   0.6     0.92    0.82     0.84     0.92     0.82
correct Ans
multidimension

Ans. by Student            GenGram  Dice  Oddgram  SumSquare  Bigram SetTrigram
methymercury         sim   0.48     0.86    0.5     0.65     0.82     0.81
correct Ans
methylmercury

Ans. by Student            GenGram  Dice  Oddgram  SumSquare  Bigram SetTrigram
stratosphare         sim   0.63     0.82    0.67     0.83      0.82     0.78
correct Ans
stratosphere

Ans. by Student            GenGram    Dice Oddgram  SumSquare  Bigram SetTrigram
chlorofluorocarbon    sim   0.47     0.91     0.5      0.61      0.88     0.89
correct Ans
chlorofluorocarbon

Ans. by Student            GenGram  Dice    Oddgram  SumSquare  Bigram SetTrigram
trypanisomic         sim   0.63     0.82    0.67     0.83      0.82     0.78
correct Ans
trypanosomic

Ans. by Student            GenGram   Dice  Oddgram  SumSquare  Bigram SetTrigram
proclaimation         sim   0.48     0.87     0.49     0.69      0.80     0.80
correct Ans
proclamation
```

Figure 5. Sample of similarity values generated by the methods

The guide of Evans [24] for absolute of *r* was employed to describe the strength of the correlation. These guides are (0.0 - 0.19) as very weak, (0.20 - 0.39) as weak, (0.40 - 0.59) as moderate, (0.60 - 0.79) as strong and (0.80 - 1.00) as very strong. A pearson's correlation was run to determine the relationship between average expert score for each student answer and similarity values of generalized n-gram, Dice, oddgram, sumsquare, bigram and set-based trigram methods. The results are as follows:

1: correlation value between generalized n-gram and average expert score is equal to 0.44
2: correlation value between Dice and average expert score is equal to 0.15
3: correlation value between oddgram and average expert score is equal to 0.59
4: correlation value between b-gram and average expert score is equal to 0.17

5: correlation value between sumsquare gram and average expert score is equal to 0.25
6: correlation value between set-based trigram and average expert score is equal to 0.19
It is of a great interest to know if there is a relationship between average expert score and similarity measures that were considered. There was a moderate correlation between average expert score for each student answer and similarity value of oddgram ($r = 0.59, N = 7,501, p < 0.001$) as well as generalized n-gram ($r = 0.44, N = 7,501, p < 0.001$). It is possible to use oddgram method to evaluate subjective examination.

9. Conclusion

This paper presented three new similarity measures of oddgram, sumsquare gram and set-based trigram. Set-based trigram and sumsquare gram returned high values of similarity and performance to price which could be useful to locate and retrieve words from word list as a group as well as relevant terms from medical database terminologies. There appears to be a very moderate positive correlation between values of oddgram and expert tutor which indicated that oddgram is suitable for the evaluation of subjective examination. The similarity values generated by the oddgram method were not exceptional better than Dice similarity values but the running times with Dice method are highly encouraging and better than generalized n-gram matching technique It was also noted that the performance of the methods was not constrained to the number of text and pattern matching due to the use of average, standard deviation, f-measure and Pearsons correlation coefficient.

10. Results from the research work

Five new methods were proposed in the research work. They are bi-n-gram, tri-n-gram, oddgram, sumsquare gram and set-based trigram. Bi-n-gram and tri-n-gram methods were proposed for the evaluation of electronic test at programming languages while oddgram method was proposed for the evaluation of electronic text at subjective examination. Sumsquare and oddgram were proposed for the retrieval of text matching from medical database, world list and case based reasoning.

Bi-n-gram and tri-n-gram methods permit to achieve a very high relatively grade results in electronic test at programming language. The number of grade produced by bi-n-gram and tri-n-gram methods are very close to the number of grades given by the experts. Set-based trigram and sumsquare gram methods are very useful in the word lists that are derived from the root of word as a group. The two methods demonstrated a great improvement in the effective retrieval of the medical terms from database by returning most of the relevant documents similar to the users' requests. The two methods also compute both pattern and matching text with high value of similarity and performance to price with low rate of processing time which satisfy the justification of the research work of closeness measurement. The methods recorded low score values of highest false match and high score values of separation which are much of the most effective in the real life of electronic test at the case based reasoning.

Oddgram method provided a moderate correlation value with the value of average expert score for each student answer which would be very useful in evaluation of electronic test at subjective examination.

References

[1] Markov, A. A., *Essai diune recherche statistique sur le text do roman*, Engene oneguine, bull. Acad imper sci. st Petersburg, Vol. 7, 1913.

[2] Spinels, D., Zaharias, R., and A., V., *Coping with plagiarism and grading load: randomized programming assignments and relflective grading*, Computer applications in engineering education, Vol. 5, No. 2, 2007, pp. 113–123.

[3] Shannon, C., *Prediction and entropy of printed English*, The bell system technical journal, Vol. 30, 1951, pp. 50–64.

[4] Zamora, E. M., Pollock, J. J., and Zamora, A., *The use of trigram for spelling error detection*, Information processing and management, Vol. 17, 1981, pp. 305–316.

[5] Burnett, J., Cooper, D., Lynch, M., Willett, P., and Wycherley, M., *Document retrieval experiments using indexing vocabularies of varying size*, Journal of documentation, Vol. 35, 1979, pp. 197–206.

[6] Trenkle, J. M. and Cavnar, W. B., *N-gram based text categorization*, In: proceedings of the symposium on document analysis and information retrieva, 1994, (University of Nevada, Los Vegas).

[7] Cheng, B. Y., Carbonell, J. G., and Klein-Seetharaman, J., *Protein classification based on text document classification techniques*, Journal of protein, Vol. 58, 2005, pp. 955–970.

[8] Nakamura, M. and Shikano, *A study of English word category prediction based on neural networks*, International conference on acoustics, speech and signal processing, Vol. 2, 1989, pp. 731–734.

[9] Tan, C. L., Sung, S. Y., Yu, Z., and Xu, Y., *Text retrieval from document images based on n-gram algorithm*, In: PRICAL workshop on text web minning, 2000.

[10] Harrison, M., *Implementation of the substring test for hashing*, Communication of the ACM, Vol. 14, 1971, pp. 777–779.

[11] Abou-Assaleh, T., Cercone, N., Keselj, V., and Sweidan, R., *N-gram based detection of new malicious code*, In: COMPSAC workshops, 2004, pp. 41–42.

[12] Abubakar, A. I. Z., *Automated grading of linear algebraic equation using n-gram method*, (Master Thesis).

[13] Barrvon-Cedeno, A. and Rosso, P., *On automatic plagiarism detection based on n-grams comparison*, Springer-Verlag Berlin Heidelberg, 2009, pp. 296–370.

[14] Fogla, P. and Lee, W., *Q-Gram matching using tree models*, IEEE transactions on knowledge and data engineering, Vol. 18, No. 4, 2006, pp. 433–447.

[15] Niewiadomski, A., *Methods for the linguistic summarization of data: application of fuzzy sets and their extensions*, Akademicka oficyna wydawnicza EXIT, Warszawa, 2008.

[16] Chaski, C., *Authorship attribution in digital evidence investigations*, International journal of digital evidence, Vol. 4, No. 1, 2005, pp. 135–143.

[17] Tversky, A. and Gati, I., *Similarity, separability and triangle inequality*, Psychological review, Vol. 89, 1982, pp. 123–154.

[18] Tversky, A., *Features of similarity*, Psychological review, Vol. 84, No. 4, 1977, pp. 327–352.

[19] Zwick, R., Carlstein, E., and Budeskco, D. V., *Measures of similarity amongst fuzzy concepts: A comparative analysis*, International journal approximate reasoning, 1987, pp. 221–242.

[20] Williams, J. and Steela, N., *Difference, distance and similarity as a basis for fuzzy decision support based on prototypical decision classes*, Fuzzy sets and systems, Vol. 131, 2002, pp. 35–46.

[21] Ismat, B. and Ashraf, S., *Fuzzy equivalence relations*, Kuwait journa science and engineering, Vol. 35, 2008, pp. 33–51.

[22] Niewiadomski, A. and Grzybowski, R., *Rozmyte miary podobienstwa tekstow w automatycznej ewaluacji testow egzaminacyjnych*, Informatyka teoretyczna i stosowana, Vol. 4, No. 6, 2004, pp. 75–79.

[23] Zhou Wei., N. R. M. and Yu, C., *A tutorial on information retrieval: Basic terms and concepts*, Journal of biomedical discovery and collaboration, Vol. 1, No. 1, 2006, pp. 1–17.

[24] Buscaldi, D., Tournier, R., Ausienac-Gillies, N., and Mothe, J., *IRIT textual similarity combing conceptual similarity with N-gram comparison method*, In: First joint conference on lexical and computational semantics, Association for computational linguistics, Montreal, Canada, 2012, pp. 41–42.