MeDict: Health Dictionary Application Using Damerau-Levenshtein Distance Algorithm

Wiwi Clarissa[1], Farica Perdana Putri[2]
[1,2] Department of Informatics, Universitas Multimedia Nusantara, Tangerang, Indonesia
[1] wiwi.clarissa@student.umn.ac.id
[2] farica@umn.ac.id

Accepted on 11 June 2020
Approved on 31 October 2020

***Abstract*—Typographical error often happens. It can occur due to mechanical errors or missed hands or fingers when typing. Someone's ignorance of how to spell correctly also can cause typographical errors. Dictionary application development has been carried out by various parties so that the searching process in the dictionary becomes more efficient. However, there is no word search optimization when the typographical error happens. Typographical errors in the searching process can result in the information sought cannot be found. The Damerau-Levenshtein Distance algorithm implemented to provide search suggestions when a typographical error occurs. This research aims to design and build a health dictionary application, MeDict, using the Damerau-Levenshtein Distance algorithm. Technology Acceptance Model (TAM) used to evaluate the application. The result is 86.2% stating strongly agree that the application can be useful and 86.9% stating strongly agree that the application can be used easily.**

***Index Terms*—Damerau-Levenshtein distance, dictionary, Technology Acceptance Model, typographical error**

I. INTRODUCTION

In the world of education, especially in the medical field, it is very important for a student majoring in medicine to understand medical terms. In the medical field, there are many terms that are difficult to understand [1]. In a book-shaped dictionary, the process of vocabulary search is still ineffective because the dictionary is large and thick, so the search process will take a long time [2]. Therefore, the development of dictionary applications has been carried out by various parties so that the search process the terms in the dictionary become more efficient. However, there was no search optimization when a typographical error occurred. Typographical errors by a user can result in the information sought can not be found.

Typographical errors can be caused by mechanical errors, such as mistyping due to finger movements. It sometimes also caused by someone's lack of knowledge about how to spell the correct word. Common mistakes made when typing include substitution, insertion, deletion, or transposition (exchanging two adjacent letters) [3].

To overcome this problem, we need a method that can be used to optimize word searching in a dictionary application. This search optimization can be done by providing search suggestions if the input word cannot be found in the dictionary.

The edit distance algorithm can be used to provide search suggestions, including the Hamming distance and Levenshtein distance algorithms [4]. Peggy has successfully implemented the Levenshtein Distance algorithm to optimize word search in Chinese - Indonesian translator applications [5]. However, research by Sutisna and Adisantoso proved that spelling correction using the Damerau-Levenshtein Distance algorithm can improve a search engine performance by 22% rather than using the Levenshtein Distance algorithm [6]. Research by Jupin, Shi, and Obradovic proved that the Damerau-Levenshtein Distance algorithm has a smaller number of errors (false positive) than the Jaro-Winkler Distance algorithm [7]. Vogler explained that the choice of a string distance algorithm depends on the problem situation being encountered. If the problem is typographical errors, then the variations of Levenshtein Distance algorithm are good, because the algorithm takes into account three or four (for Damerau-Levenshtein Distance) types of typing errors that usually occur [8].

Based on the previous researches, this health dictionary application called MeDict uses the Damerau-Levenshtein Distance algorithm to optimize word searching. The Damerau-Levenshtein Distance algorithm will be used to correct typographical errors by giving word suggestions that have similarities according to the Damerau-Levenshtein Distance calculations.

II. LITERATURE REVIEW

A. *Damerau-Levenshtein Distance*

The Damerau-Levenshtein Distance algorithm was developed by Frederick J. Damerau. Damerau-

Levenshtein Distance is a measurement (metric) produced through the calculation of the number of differences found in two strings. The Damerau-Levenshtein Distance algorithm determines the minimum number of operations needed to convert one string into another string.

Damerau-Levenshtein Distance algorithm is a development of the Levenshtein Distance algorithm. Damerau extended Levenshtein distance to also detect transposition errors and treat them as one edit operation [7]. Therefore Damerau-Levenshtein calculates the minimum insertion, deletion, substitution, and transposition operations to convert one word into another. Damerau stated that about 80% of typographical errors were the result of all four operations.

The pseudocode of the Damerau-Levenshtein Distance algorithm can be seen in Table 1.

TABLE I. PSEUDOCODE OF DAMERAU-LEVENSHTEIN DISTANCE ALGORITHM [9]

Damerau-Levenshtein Distance Algorithm

```
function dameraulevenshteinDistance(input s : array[1..m] of char, input t : array[1..n] of char) ◊ integer {function to compute Damerau-Levenshtein distance between two strings using Damerau-Levenshtein algorithm

DECLARATION
i, j : integer cost : integer d : array [0..m][0..n] of integer

ALGORITHM
for i ← 1 to m do { source prefixes initialization }
  d[i][0] ← i
endfor
for j ← 1 to n do { target prefixes initialization }
  d[0][j] ← j
endfor
{ using Damerau-Levenshtein Algorithm to check } for i ← 1 to n do
  for j ← 1 to m do
    if (s[i] == t[j]) then
      cost ← 0
    else
      cost ← 1
    endif
    d[i][j] ← minimum (
      d[i-1][j] + 1, { deletion }
      d[i][j-1] + 1, { insertion }
      d[i-1][j-1] + cost { substitution }
    )
    if (i > 1 and j > 1 and s[i] == t[j-1]
    and s[j-1] == t[i]) then
      d[i][j] ⇓ minimum (
        d[i][j],
        d[i-2][j-2] + cost { transposition }
      )
    endif
  endfor
endfor
→ d[m][n] { return results }
```

B. *Filter and Verify Method*

In the 90s, the "filter and verify" method was introduced to reduce data comparisons in the calculation of edit distance. Research on this method is still very active. Filters can make the system more efficient by removing unnecessary comparisons. One of the most common methods is length filtering, where the difference in the length of the two strings *s* and *t* must not be greater than *k* [7]. The algorithm of length filtering can be seen in Table 2.

TABLE II. LENGTH FILTERING ALGORITHM [7]

Length Filtering Algorithm

```
Algorithm: LengthFilter(s, t)
Input: s, t: strings of characters
Output: Boolean
Begin
  if abs(|s| - |t|) > k : return FALSE
  else: return TRUE
  end-if
end
```

C. *Technology Acceptance Model (TAM)*

The Technology Acceptance Model (TAM) was introduced by Fred D. Davis in 1989 as an instrument for predicting the possibility of new technology being adopted in a group [10].

The Technology Acceptance Model can be illustrated in Fig. 1. According to this model, the user's attitude towards the use of a given system is considered to be the major determinant of whether he uses it or not. Attitudes toward use are influenced by two variables: perceived usefulness and perceived ease of use. Perceived usefulness is the degree to which an individual believes that using a particular system will improve the performance of his work. Perceived ease of use is the degree to which an individual believes that using a particular system will be free of physical and mental effort. Perceived usefulness is also influenced by perceived ease of use because a system that is easier to use will result in increased job performance. Design features directly influence perceived usefulness and perceived ease of use [11].

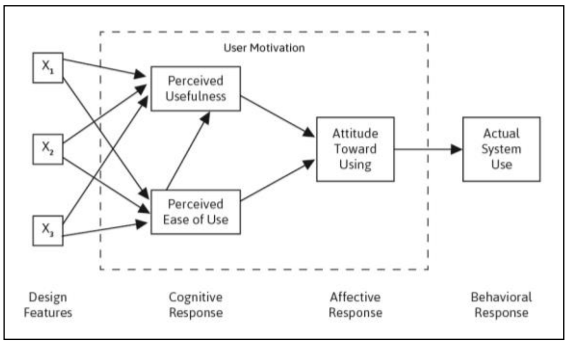


Fig. 1. First phase of test setup 1 of ADS-B signal quality testing with receiver inside the walls

Initially, Davis used 14 indicators (initial scale items) in measuring perceived usefulness and perceived ease of use. But after several trials, the results obtained in the form of 6 measurement indicators are better and more practical. Table 3 is a measurement indicator for the variables of perceived usefulness and perceived ease of use.

TABLE III. INDICATORS OF PERCEIVED USEFULNESS AND PERCEIVED EASE OF USE [12]

Scale Items
Usefulness
1. Work More Quickly
2. Job Performance
3. Increase Productivity
4. Effectiveness
5. Makes Job Easier
6. Useful
Ease of Use
1. Easy to Learn
2. Controllable
3. Clear & Understandable
4. Flexible
5. Easy to Become Skillful
6. Easy to Use

III. EXPERIMENTAL RESULTS

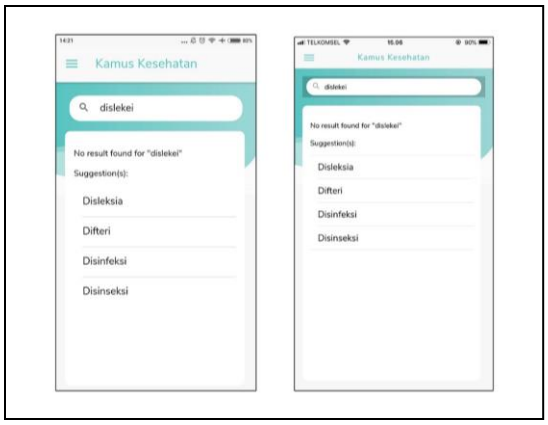
Damerau-Levenshtein Distance algorithm will be evaluated by comparing the results of manual calculation with the results of the calculation of edit distance by the application. Following is the scenario of testing the Damerau-Levenshtein Distance algorithm. Table 4 is a sample of data entered by the user.

TABLE IV. USER SAMPLE DATA

Words typed	Words supposed to be
dislekei	disleksia
neuorablastona	neuroblastoma
influnea	influenza
frotifikasi	fortifikasi
black water feaver	blackwater fever

The application will calculate the edit distance value using the Damerau-Levenshtein Distance algorithm and provide a list of word suggestions that are similar to words entered by the user. Fig. 2 shows the search result for the word "dislekei" in the health dictionary application. Based on Fig. 2 it can be seen that the application can provide search suggestions when typographical error occurs.

The tolerance value used in this application is 50%, meaning that the application will only display word suggestions that have an edit distance that is less than or equal to 50% of the number of letters entered by the user. The word "dislekei" has 8 letters, meaning the application will display word suggestions that have an edit distance value of less than or equal to 4.



No result found for "dislekei"

Suggestion(s):

Disleksia

Difteri

Disinfeksi

Disinseksi

Fig. 2. User's Search Result

Table 5 is the manual calculation of the Damerau-Levenshtein Distance algorithm. The last cell colored in green shows the edit distance value between the words "dislekei" and "disleksia", which is 2.

TABLE V. DAMERAU-LEVENSHTEIN DISTANCE CALCULATION

		d	i	s	l	e	k	s	i	a
	0	1	2	3	4	5	6	7	8	9
d	1	0	1	2	3	4	5	6	7	8
i	2	1	0	1	2	3	4	5	6	7
s	3	2	1	0	1	2	3	4	5	6
l	4	3	2	1	0	1	2	3	4	5
e	5	4	3	2	1	0	1	2	3	4
k	6	5	4	3	2	1	0	1	2	3
e	7	6	5	4	3	2	1	1	2	3
i	8	7	6	5	4	3	2	2	1	2

Fig. 3 is the result of the Damerau-Levenshtein Distance calculation by the application. It shows that the result of the edit distance is 2. Based on Table 5 and Fig. 3, it can be seen that the calculation result in the health dictionary application equals to the result of the manual calculation.

```
distance = d[8][9] = 2
```

Fig. 3. Damerau-Levenshtein Distance Calculation Result by Application

Application acceptance testing was also conducted in this study. The method used in testing application acceptance is based on the Technology Acceptance Model (TAM) by distributing questionnaires. The sampling technique used was purposive sampling technique. Therefore, the questionnaire was given to 35 respondents related to the medical field, namely medical students and nursing students to get an assessment of this health dictionary application. Questionnaire questions are divided into two parts: perceived usefulness and perceived ease of use.

Table 6 is the answer to the questionnaire for the perceived usefulness variable. Based on the calculation of the total score of the perceived usefulness variable, it can be concluded that 86.2% of users strongly agree that this health dictionary application can improve work performance and be useful.

TABLE VI. PERCEIVED USEFULNESS QUESTIONNAIRE RESULT

Questions	1	2	3	4	5
The Medict application speeds up my work in finding the meaning of a medical term	0	1	3	15	16
Using the Medict application can improve my work performance	0	0	4	17	14
In my opinion, using Medict application can increase my productivity	0	1	4	18	12
In my opinion, the use of Medict application can help me search the meaning of a medical term effectively	0	1	5	10	19
In my opinion, the Medict application can facilitate me in finding the meaning of a medical term	0	0	4	14	17
Overall, the Medict application is useful	0	0	4	14	17

Table 7 is the answer to the questionnaire for the variable perceived ease of use. Based on the calculation of the total score of the ease of use variable, it can be concluded that 86.9% of users strongly agree that the health dictionary application is easy to use.

TABLE VII. PERCEIVED EASE OF USE QUESTIONNAIRE RESULT

Questions	1	2	3	4	5
In my opinion, Medict application is easy to learn	0	1	3	18	13
In my opinion, Medict application can be run according to its function	0	0	3	18	14
In my opinion, Medict application is clear and understandable	0	1	4	11	19
My interaction with Medict application is easy for me to understand	0	0	2	16	17
I can easily familiarize myself with every feature in Medict application	0	1	4	14	16
Overall, Medict application is easy to use	0	0	2	16	17

IV. CONCLUSION

The health dictionary application has been successfully designed and built using the Damerau-Levenshtein Distance algorithm. The application is built based on mobile which can be used on devices with the Android and iOS operating systems. The programming language used to build this application is Typescript using the Ionic framework. The health dictionary application can provide search suggestions with the Damerau-Levenshtein Distance algorithm calculation if there are typographical errors. Search suggestions given to users are sorted from the lowest to highest edit distance values. The implementation of length filtering method also works fine to reduce the comparison of words that are not needed.

This application has been evaluated by 35 respondents using the Technology Acceptance Model (TAM) and obtained a result of 86.2% states strongly agree that the application can be useful (perceived usefulness) and 86.9% states strongly agree that the application can be easily used (perceived ease of use).

REFERENCES

[1] Lestari, C. P., Hasibuan, N. A., and Ginting, G. L. “Perancangan Aplikasi Kamus Istilah Medis Berbasis Android dengan Algoritma Boyer Moore”. Jurnal INFOTEK, vol. II, no. 3, June 2016, pp.28-32.

[2] Pratiwi, H., Arfyanti, I., and Kurniawan, D. “Implementasi Algoritma Brute Force dalam Aplikasi Kamus Istilah Kesehatan”. Jurnal Ilmiah Teknologi Informasi Terapan, vol. II, no. 2, 2016, pp.119-125.

[3] Dwitiyastuti, R. N., Muttaqin, A., and Aswin, M. “Pengoreksi Kesalahan Ejaan Bahasa Indonesia Menggunakan Metode Levenshtein Distance”. Jurnal Mahasiswa Teknik Elektro Universitas Brawijaya, 2013.

[4] Budiman, A., Dennis G., Seng Hansun. “Implementasi Algoritma Hamming Distance dan Brute Force dalam Mendeteksi Kemiripan Source Code Bahasa Pemrograman C”. ULTIMATICS, vol. 8, no. 2, 2016.

[5] Peggy and Hansun, S. “Optimasi Pencarian Kata pada Aplikasi Penerjemah Bahasa Mandarin – Indonesia Berbasis Android dengan Algoritma Levenshtein Distance”. ULTIMA Computing, vol. VII, no. 1, 2015.

[6] Sutisna, U., and Adisantoso, J. “Koreksi Ejaan Query Bahasa Indonesia Menggunakan Algoritme Damerau Levenshtein”. Jurnal Ilmiah Ilmu Komputer, vol. 15, no. 2, 2010.

[7] Jupin, J., Shi, J. Y., and Obradovic, Z. “Understanding Cloud Data Using Approximate String Matching and Edit Distance”. 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, 2012, pp. 1234-1243.

[8] Vogler, R. 2013. Comparison of String Distance Algorithms. [Online]. Available on: https://www.joyofdata.de/blog/comparison-of-string-distancealgorithms.

[9] Setiadi, I. “Damerau-Levenshtein Algorithm and Bayes Theorem for Spell Checker Optimization”, 2013.

[10] Tang, D. P., and Chen, L. J. “A Review of the Evolution of Research on Information”. 2011 International Conference on Business Management and Electronic Information, 2011, pp. 588-591.

[11] Davis, F. D. “A Technology Acceptance Model for Empirically Testing New End-User Information Systems : Theory and Results”, 1985.

[12] Davis, F. D. “Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology”. MIS Quarterly, vol. 13, no. 3, 1989, pp. 319-340.