Damerau levenshtein distance for indonesian spelling correction

Puji Santoso[a,1,*], Pundhi Yuliawati, Ridwan Shalahuddin, Aji Prasetya Wibawa[b,2]

[ab] Jurusan Teknik Elektro, Universitas Negeri Malang
[1]pujiosotnas@gmail.com, [2]pe.yhulia@gmail.com, [3]ridhwan102@gmail.com, [4]aji.prasetya.ft@um.ac.id
** corresponding author*

ABSTRACT

Word correction used to find an incorrect word in writing. Levenshtein distance is one of the algorithms for correcting typing error. It is an algorithm that calculates a difference between two strings. The operations that used to the calculation are insert, delete, and substitution. However, this algorithm has a disadvantage that it cannot overcome two switched letters in the same word. The algorithm that can solve those issues is a Damerau Levenshtein. This research aims to analyze a Damerau Levenshtein algorithm used for correcting Indonesian spelling. The dataset in this research consists of two fairy tale stories with 1266 words and 100 typing errors. From these two algorithms, the accuracy is up to 73% on Levenshtein distance and 75% on Damerau Levenshtein.

Keywords:
Data Mining
Damerau Levenshtein
Distance
Spell Checker

I. Introduction

Levenshtein distance is a matrix measurement obtained from the calculation of two strings [1]. Two strings on Levenshtein distance represent the number of minimum changes required to substitute a string with another string [2]. The operations used in Levenshtein distance are insert, delete, and substitution [3]. However, this algorithm has a disadvantage that if changes in two letters should be changing these two letters. Therefore, it requires another algorithm that can solve the problem that is damerau Levenshtein distance.

Damerau Levenshtein distance is an improvement of the Levenshtein distance algorithm. In this algorithm, there is four required minimum operation to change a string into another string. These operations are insertion, deletion, substitution, and additional transposition operation [4].

This research aims to examine a comparison between Levenshtein distance and the damerau Levenshtein distance algorithm in a fairy tale story. It expected to provide an accuracy level of these two algorithms in correcting Indonesian spelling. A comparison result will provide information on a better algorithm for Indonesian spelling.

II. Method

A. Reseach Dataset

T A dataset used are two fairy tale stories from ceritadongengrakyat.com. These stories consist of 1266 words. There are 100 typing errors in these two stories. The stories will adjust from several problem scenarios to test the program.

B. Data Processing

The preprocessing stage is performed before conducting the trial. This stage aims to remove a number and read signs in a story. This stage has done so while testing performed, there is no number and read the sign that can affect the result. Besides, the dictionary used is limited to save an initial word, so it s requires additional affix words. The examples of affix words are "*membantu*", "*melakukan*", "*menolong*", "*membuat*", and so on.

C. Spelling Error

Spelling errors occur if the author's words do not appear suitable to KBBI (Kamus Besar Bahasa Indonesia). Spelling errors are caused by several things, such as ignorance in writing, errors during

DOI: 10.26555/jifo.v13i2.a15698

writing or typing, and errors in the machine during storage. Spelling mistakes that usually occur are caused by four things as follows:

1. Substitution of one letter
   Example: the word 'Wayang' becomes 'Wayank'.
2. Insertion of one letter
   Example: The word 'Wayang' becomes 'Wwayang'.
3. The omission of one letter
   Example: The word 'Gurita' becomes 'Urita'.
4. Exchange two adjacent letters
   Example: The word 'Wayang' becomes 'Waynag'.

D. *Damerau Levenhstein Algorithm*

The damerau Levenshtein is an improvement of the Levenshtein distance algorithm. In this algorithm, the minimum number of operations needed to convert one string into another string is calculated. The processes used on the Levenshtein distance algorithm are insertion, deletion, substitution. While, in the damerau Levenshtein distance algorithm, the operation used is almost the same as Levenshtein distance, but with the addition of the transposition operation between two characters [5]. Damerau Levenshtein does not distinguish between these four operations. The developed algorithmic process is compatible with the percentage of 80% of all errors in personal writing. Errors usually occur in the loss of letter characters, excess character letters, or error sequence letters of two different letter characters [6]. Examples application of the operations used in the Damerau Levenshtein Distance algorithm are as follows:

1. Insertion is an operation by inserting characters at a particular index to match the source string and the target string.
2. Deletion is an operation by deleting characters at a specific index to match the source string and the target string.
3. Substitution is an operation by replacing characters at a particular index to match the source string and the target string.
4. Transposition is an operation by swapping characters at a particular index to match the source string and the target string.

E. *Calculation of Damerau Levenshtein Algorithm*

Calculations on damerau Levenshtein are done by calculating the edit distance illustrated in Table 1 with the wrong word MALAGN and the target word MALANG. The estimate of the value of edit distance is obtained from the meeting of each row and column. The calculation starts at the index position of the first and last row and column. The results can be known after calculating until the end of the column, which will be the value of the edit distance. The following is the edit distance value, which is calculated at the end of the row and column.

Table 1. Example of calculation of edit distance value in damerau Levenshtein

S/T		M	A	L	A	N	G
	0	1	2	3	4	5	6
M	1	0	1	2	3	4	5
A	2	1	0	1	2	3	4
L	3	2	1	0	1	2	3
A	4	3	2	1	0	1	2
N	5	4	3	2	1	1	1
G	6	5	4	3	2	1	1

The process in Levenshtein's damerau algorithm is as follows [7]:

1. Initialize n for the character length of s and m for the character length of t.
   If n = 0 or m = 0, then return value in the form of edit distance to the formula:
   edit_rate = max (n, m)
   then skip to step 7.
2. Create a matrix d of m + 1 row and n + 1 column.

3. The first row contains 0..n and fills the first column with 0..m.
4. Check each character from s to t
   If s[i] = t[j] then cost = 0.
   If s[i] ≠ t[j] then cost = 1.
5. Fill in the values of each cell d [i, j] row by row with:
   d[i, j] = min(x, y, z)
   explanation:
   d[i, j] : cell that shows the column meeting with row i in matrix d.

   x : the value that is in the upper cell from the current cell position plus 1 (one) or can be formulated: *x = d[i - 1, j]+1*

   y : the value in the cell to the left of the current cell position plus 1 (one) or can be formulated: *y = d[i, j - 1] + 1*

   z : the value contained in the upper cell from the left cell now (northwest) plus the cost value and can be formulated: *z = d[i – 1, j – 1] + cost*

   If i > 1 and j > 1 and s[i] = t[j - 1] and s[i - 1] = t[j] It means after the two words are compared there are characters that can be transposed, then fill in the values of cells d [i, j] with the following formula:
   d[i, j] = min(d[i, j], d[i - 2, j - 2] + cost)
6. Next is to determine the calculated edit distance, which can be found in cells d [n, m] in the right corner of the last row.
7. End

F. *Words Recommendation and Accuracy*

The results of the program suggest that the detected word is incorrect. Suggestions appear incorrect words. The word suggestion given will refer to the dictionary used in the program database. Accuracy calculation is done to count the number of wrong words that can be corrected after that divided by the wrong words multiplied by 100%. The formula to calculate the accuracy (1). x is the number of incorrect words that can be corrected, and n is the number of incorrect words.

$$Accuracy = \frac{x}{n} x\ 100\% \qquad (1)$$

G. *Flowchart Program*

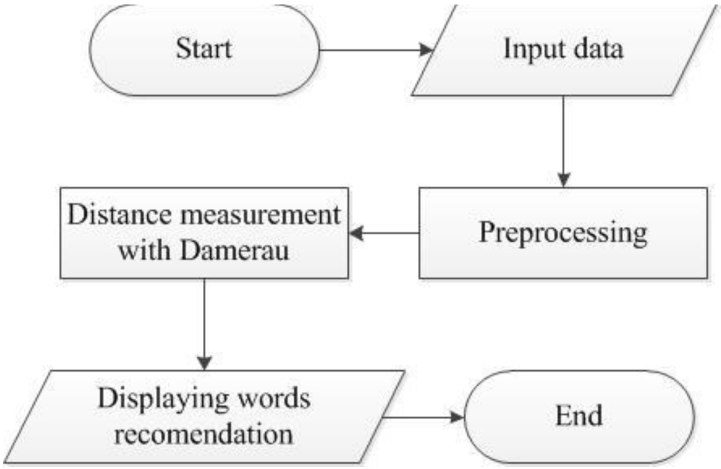
Start
Input data
Preprocessing
Distance measurement with Damerau
Displaying words recomendation
End

Fig. 1. Flowchart program

III. Result and Discussion

The test performed on two fairytale stories consisting of 1266 words and 100 spelling errors. The method used to test is Levenshtein distance and damerau Levenshtein. The results obtained from the two methods are differences in the suggested words, as in the example of Table 2.

Table 2. Example of incorrect words

Incorrect words	Levenshtein Distance	Damerau Levenshtein
Abtang	abang	batang abang
Hatti	hatta hati	hati hatta
Sedij	sedia sedih	sedih sedia
Sdang	slang siang sadang sang sidang soang udang dang sedang	sedang siang udang adang sang sidang dang slang
Kaal	kail kadal kial kapal	kapal kial kala kanal
Makhulk	-	makhluk
Bajnir	banir	banir banjir
Degnan	degan	degan dengan

Table 2 shows the test results using Levenshtein distance and damerau Levenshtein. The results show that damerau Levenshtein has better results in suggesting the wrong word. As in Table 2, some words cannot be suggested using the Levenshtein distance method. For example, words that cannot be suggested by Levenshtein distance such as “abtang” suggested “abang”, and “bajnir” suggested “banir”. While, in damerau Levenshtein, the word can be suggested. Some words do not have a suggestion on Levenshtein distance, which is on the word “makhulk”, because the way Levenshtein distance works cannot move two interchangeable letters. Whereas in damerau Levenshtein, the problem can be overcome by transposing the two exchanged letters. Meanwhile, the testing program has a difference in calculation time between Levenshtein distance and damerau Levenshtein, as in Table 3.

Table 3. Testing time

No.	Number of letters	Incorrect letter	Levenshtein Distance(s)	Damerau Levenshtein(s)
1	132	45	27.30	33.18
2	269	57	35.76	44.21
3	254	48	34.35	39.36
4	403	64	40.84	49.43
5	523	81	46.71	51.85
6	496	70	45.10	49.47
7	413	83	50.26	55.58
8	775	172	103.30	118.05
9	2.683	82	53.31	60.74
10	3.392	110	64.56	72.58

Table 3 shows the time difference from the calculation of the program between the two algorithms. Based on the results, in terms of time, Levenshtein distance has a better time than damerau Levenshtein. The time required for damerau Levenshtein's calculations is relatively longer. One that affects the calculation time is the number of letters in the test data and the number of letters in the incorrect word. The average required time for a Levenshtein distance algorithm when calculating the test data with some letters under 2000 is 52.14 seconds.

Meanwhile, the average calculation of the Levenshtein noise algorithm is 55.49 seconds. The average required time for a Levenshtein distance algorithm when calculating the test data with some letters above 2000 is 59.13 seconds, and the Damerau Levenshtein algorithm is 66.67 seconds. Besides, the calculation of the longest time between the two methods is the number of incorrect 172 letters.

Moreover, damerau Levenshtein has the disadvantage of cannot to correct two words that have no spaces. Thus, some mistakes cannot be corrected by damerau Levenshtein, as in Table 4. Furthermore, the testing result of damerau Levenshtein shown in Table 5.

Table 4. Word That Cannot Be Suggested

Incorrect words	Suggested words
putriraja	-
suatuhari	-
adalahtitisan	-
binatangdan	-
anjingitu	-
tidakpercaya	-
tanpasadar	-
yangtampan	-
seoranggadis	-
bekasluka	-

Table 5. Testing result

Algorithm	Spelling Error	Can be corrected	Cannot be corrected	Accuracy
Levenshtein Distance	100	75	25	75%
Damerau Levenshtein	100	73	27	73%

Based on Table 5, the Levenshtein distance algorithm can correct the error results as many as 73 out of a total of 100 errors and 27 errors cannot be corrected. Damerau Levenshtein's algorithm can correct 75 out of 100 errors, and 25 errors cannot be corrected. The damerau Levenshtein algorithm has better results than the Levenshtein distance algorithm by increasing the accuracy results from 73% to 75%. Errors cannot be corrected because the two algorithm's methods cannot correct words that do not have space or two words attached.

IV. Conclusion

The implementation of the damerau Levenshtein algorithm to correct spelling in children's fairy tales story has better results than the Levenshtein distance algorithm. The accuracy of the test is the Levenshtein distance algorithm by 73% and the damerau Levenshtein algorithm by 75%. Besides, damerau Levenshtein has better word suggestions compared to Levenshtein distance. However, in this research, some weaknesses must be developed again by the next researcher. Some errors that cannot be resolved are correcting two words that are attached or do not have spaces and too many word suggestions. Moreover, the processing time required by the damerau Levenshtein algorithm tends to be longer than the Levenshtein distance.

References

[1] Z. Su, B. Ahn, K. Eom, M. Kang, J. Kim, and M. Kim, “Plagiarism Detection Using the Levenshtein Distance and Smith-Waterman Algorithm,” pp. 0–3, 2008.

[2] D. Nofriansyah, S. N. Arief, and B. Anwar, “Optimization of Levenshtein Technique and Intrusion Detection System Method to Overcome in the Middle Attack From Intruder on Based Network TPC/IP,” pp. 203–212, 1978.

[3] F. Ahmad, “Learning a Spelling Error Model from Search Query Logs,” no. October, pp. 955–962, 2005.

[4] E. Brill and R. C. Moore, “An Improved Error Model for Noisy Channel Spelling Correction,” no. Kukich 1992, 2000.

[5] J. Jupin, J. Y. Shi, and Z. Obradovic, “Understanding Cloud Data Using Approximate String Matching and Edit Distance,” pp. 1234–1243, 2013, doi: 10.1109/SC.Companion.2012.149.

[6] D. Q. Thang, “Determining restricted Damerau-Levenshtein edit- distance of two languages by extended automata,” 2010.

[7] R. Gabrys, E. Yaakobi, and O. Milenkovic, “Codes in the Damerau Distance for DNA Storage,” no. January, 2016.