Mixed Heuristic Algorithm As String Matching For Search Document

RD Sari[1] , R Rahmadani[2] , TTA Putri[3]
{ressy@unimed.ac.id[1], renirahmadani@unimed.ac.id[2], tansatrisna@unimed.ac.id[3]}

[1,2,3] PTIK-FT, Universitas Negeri Medan, Indonesia

Abstract. The computer provides document search based on the document file title. So this makes it difficult for users to find documents, if the user forgets the file title of the document. String matching algorithm is the basic component for data searching. Search engine requires an algorithm that can work quickly and can sort the documents according to the level of compatibility. One of the algorithms that match is the Mixed Heuristic. These algorithms perform a search pattern or query not just against a word, but can be a sentence of more than one word. In addition, these algorithms also perform ranking of relevant documents. This paper shows the analysis of the level of accuracy using precision and recall of the results given by search engines by using the Mixed Heuristic algorithms for string matching, and the analysis of documents from the results given by search engines.

Keywords: search engine, string matching, mixed heuristic

1. Introduction

It is very often for people to use computers to store documents. The computer provides document search based on the document file title. So this makes it difficult for users to find documents, if the user forgets the file title of the document. For this reason, a search engine is needed that can search documents quickly and be sorted according to the level of compatibility. Mixed Heuristic Algorithm is an algorithm that adopts the well-known Heuristic algorithm, which is Brute Force. Mixed Heuristic Algorithm does a search pattern or query not only for 1 word, but can be a sentence that is more than 1 word. A string matching pattern shifts the pattern by 1 word, if a match has not been found.

The implementation of Mixed Heuristic algorithm as string matching in document search will perform the calculation and analysis of the accuracy with the precision and recall parameters based on the number of datasets. And also analyzed the results of ranking given by the system. Mixed Heuristic Algorithm is part of the system searching phase. The results given will be tested on several users of the system so that the value obtained is the average value of the test results. Analysis of ranking results is also carried out testing of several users.

Algoritma Mixed Heuristic

According to Atif Agha Hassan, the Mixed Heurustic algorithm is the result of the adoption of an algorithm using the Heuristic technique, the Brute Force. Mixed Heuristic Algorithm does a search pattern or query not only for 1 word, but can be a sentence that is more than 1 word. A string matching pattern shifts the pattern by 1 word, if a match has not been found. This

ACEIVE 2019, November 16, Medan, Indonesia

DOI 10.4108/eai.16-11-2019.2293280

algorithm first works by utilizing exact String Matching and if a solution is not found to be exact or 100%, then this algorithm will estimate the matching string by calculating the probability of success and the most feasible solution. The match percentage is saved for use when ranking [1].

Implementation Mixed Heuristic Algorithm.

In doing string matching, the Mixed Heuristic algorithm has a pattern. For the first pattern, the Mixed Heuristic algorithm relies on matching strings exactly as given keywords. And if an exact solution is not found, then the subsequent matching uses a matching estimate string to calculate the probability of success and get the most feasible solution [1]. Using this estimated string is due to user behavior that changes the search keyword if a match is not found with the keyword used before, such as minimizing the search string again and again or multiple times adding a string to the keyword [1].

If a new user uses a search engine and doesn't know the search strategy, it will be difficult to find results that are suitable for the user. For example can not find what is needed or can not find the right results. By using this Mixed Heuristic algorithm, it will be given matching patterns with keywords that come from the user. With this algorithm, users do not need to change keywords if a match is not found. Because this algorithm has provided combinations of keywords given by the user.

Given an example of a Mixed Heuristic algorithm pattern so that it can be seen the shape of the pattern given by the Mixed Heuristic algorithm.

Example: n = 5, n is the number of words in the search string

a b c d e	1 1 1 1 1
b c d e a	1 1 1 1 1
c d e a b	1 1 1 1 1
d e a b c	1 1 1 1 1
e a b c d	1 1 1 1 1
a b c d	1 1 1 1 0
a b c e	1 1 1 0 1
a b d e	1 1 0 1 1
a c d e	1 0 1 1 1
b c d e	0 1 1 1 1
a b c	1 1 1 0 0
a b e	1 1 0 0 1
a d e	1 0 0 1 1
c d e	0 0 1 1 1
b c d	0 1 1 1 0
a b	1 1 0 0 0
a e	1 0 0 0 1
d e	0 0 0 1 1
c d	0 0 1 1 0
b c	0 1 1 0 0
a	1 0 0 0 0
e	0 0 0 0 1
d	0 0 0 1 0
c	0 0 1 0 0
b	0 1 0 0 0

2. Research Method

The system built is a system that implements Mixed Heuristic algorithm in matching strings and is applied in searching indexes. The input from the system is an Indonesian language news document downloaded from www.kompas.com. This document was previously saved as * .txt. Figure 1 below is a picture of the stages of building a system.

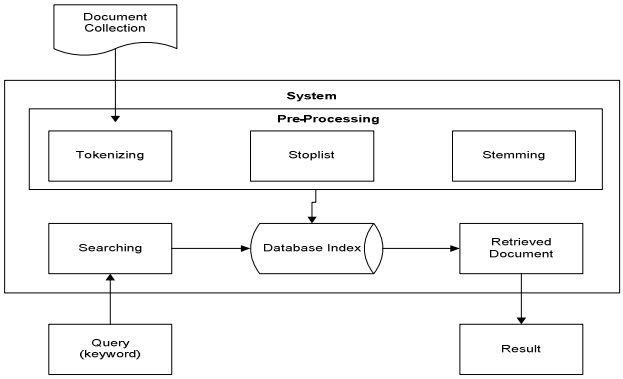

Fig 1. Stages of Development System

The explanation of Figure 1 is as follows:

1) Searching: This process of searching a document by the user, starts with the user entering a query in the search engine. At this stage, the Indexing process is performed on queries entered by the user. The results of the indexing in the form of a collection of terms, will be matched with the query entered by the user [2].
2) Pre-processing: Processing is done in this stage to convert text into usable data [3]. In this process include:
   a. Tokenizing: the stage of cutting the input string based on each word that makes it up. All words in a document will be broken down according to the words of the compiler. At this stage a case folding process is also carried out, which changes all the letters in the document into lowercase letters. Only letters atau A-Z ',' a-z ', and numbers' 0-9 'are accepted [4].
   b. Stoplist: Stoplist is the process of deleting words that are very often displayed in documents such as: and, or, not and so on [4].
   c. Stemming: This stage performs the process of returning various word formations into the same representation. Stem (root word) is the part of the word that remains after the affix is removed (prefix and suffix) [4].

3. System Design

In the system to be built, functional requirements that must be met are:

1. Input data is an Indonesian language news document with * .txt format done offline.
2. Doing the process of cutting the words that compose the sentence in the document into tokens (tokenizing stage), the process of removing common and unimportant words

(stoplist) and returning the words to their basic form (stemming stage). The stemming algorithm uses the Porter algorithm which is adapted to Indonesian [5].
3. Do the indexing process.
4. Perform the searching process using the Mixed Heuristic algorithm.
5. See the level of relevance of documents produced by Mixed Heuristic algorithm based on document ranking.
6. Seeing the searching performance of relevance values, namely precission and recall.

Analysis System Test

System testing uses the results given by the system, which then results will be given the value of precision, recall. Tests are carried out by 7 users to get the value of the number of relevant documents retrieved. The average value is the value used for the calculation of precision and recall [6].

$$\text{precision} = \frac{\text{Number of relevant documents retrieved}}{\text{The number of documents retrieved}} \quad (1)$$

$$\textit{recall} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of relevant documents}} \quad (2)$$

3. Results adn Discussion

3.1 Analysis of Testing Results on Number of Documents Viewed from Precision and Recall

After testing with a predetermined scenario, the results are obtained that the greater the document, the greater the chance for users to get relevant data. For more details, Figure 2 is a graph that shows the relationship of the number of documents and the value of precision, recall.

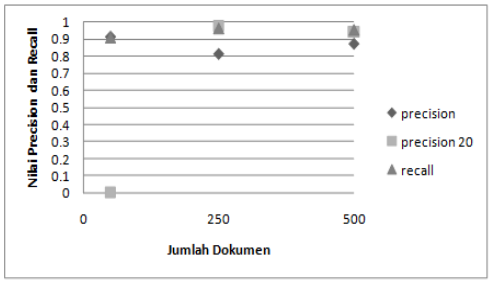

Fig 2. Graph of Test Results for Document Number

3.2 Analysis of Test Results for Term Amounts in Queries

The optimal results for this test will be achieved with the condition that all possible term combinations are retrieved by the system and the results are relevant according to the user,

because the system with the Mixed Heuristic algorithm works by matching strings for all query combinations. Can be seen from the graph, the optimal results achieved for examples 1 and 3.

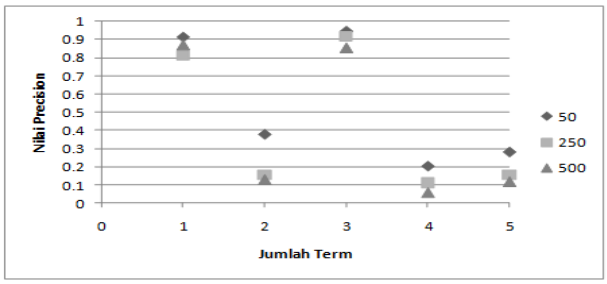

Fig 3. graph of the number of terms to the value of precision

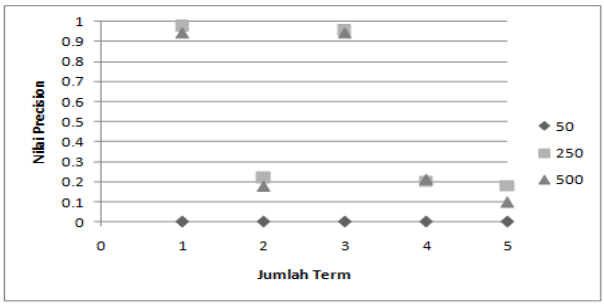

Fig 4. Graph of the number of terms to the value of precision 20

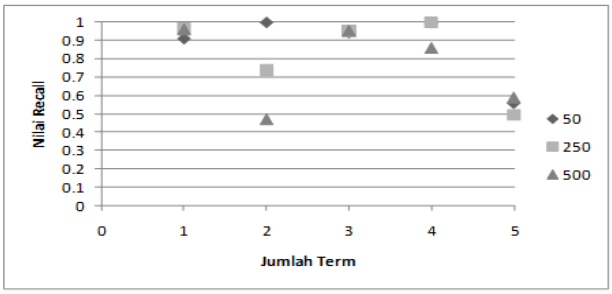

Fig 5. Graph of the number of terms to the recall value

3.3 Analysis of Test Results of Term Effect with Stopword on Queries

The more the number of terms, the value of precision and recall tends to fall because the possibility of finding documents with sentences that match or equal to the query is increasingly difficult. The combination of terms given by the system more and more, the possibility of relevant documents is also less because the system provides documents that come from a combination of one or two query terms. This can be seen from the graph below.

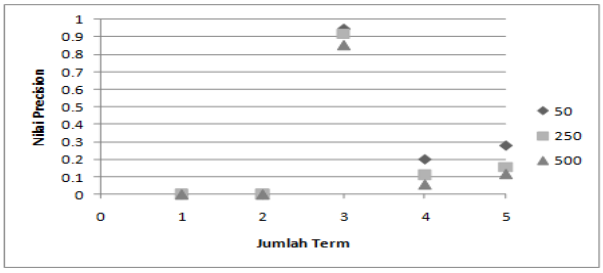

Fig 6. Graph of precision values to terms with stopwords

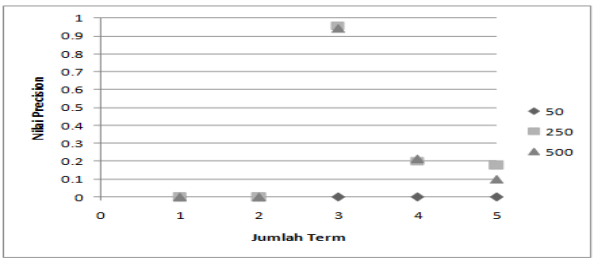

Fig 7. Graph value of precision 20 to term with stopword

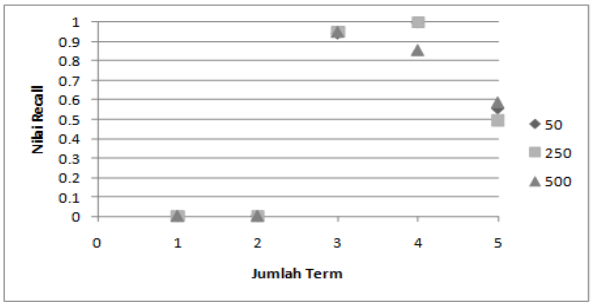

Fig 8. Graph of recall value of terms with stopword

3.4 Analysis of Test Results of Term Effects without Stopword on Queries

For the use of terms without a stopword in the query, the preprocessing process will not change the length of the query term. From the three graphs below, we can see that the values of

precision and recall tend to go down because the possibility of finding documents with the same sentence or the same as the query becomes more difficult.

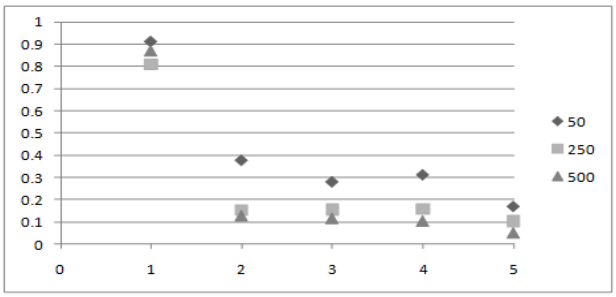

Fig 9. Value of precision term without stopword

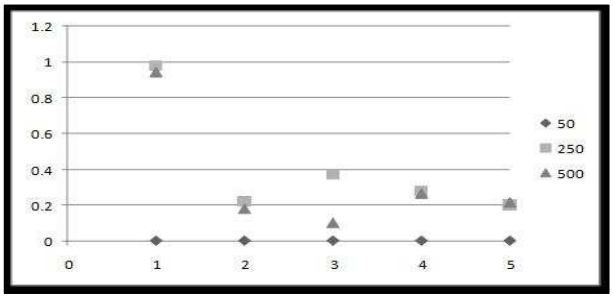

Fig 10. The value of precision is 20 terms without stop word

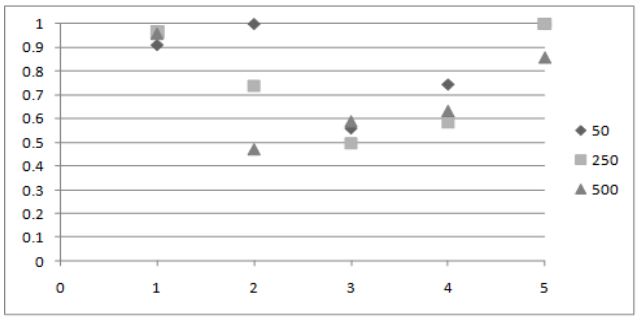

Fig 11. The value of recall term without stop word

5.5 Analysis of System Ranking Test Results

This test is done because the workings of the Mixed Heuristic algorithm not only provide relevant document results, but also provide value to the document. This value is used for ranking documents when the system gives results to the user. The following document ranking formula is used:

$$\text{Ranking value formula} = \frac{\text{Incorrect number of documents retrieved}}{\text{The number of relevant documents retrieved}} \quad (3)$$

Following are the test results for ranking the system using 250 document numbers and the number of terms given in the following table:

Table 1. System ranking results

Jumlah Term	Jumlah Dokumen ter-*retrieve*	Jumlah Relevan ter-*retrieve*	Urutan Perangkingan yang Relevan (20 teratas)	Nilai persentase kesalahan (20 teratas
1	25	21	1-20	0%
2	29	6	7, 12, 13, 16, 17, 19	0%
3	28	27	1-20	0%
4	36	4	1,2,3,4	0%
5	48	15	1,2,3,8,17,18	60%

From table 1, we can see the top 20 ranking has a low percentage for the error rate of the ranking retrieved by the system. With an average percentage of 12%.

4. Conclusion

1) Mixed Heuritic Algorithm can be applied for document search, marked with a recall value above 0.8 from a scale of 0-1 for the five tests.
2) Mixed Heuristic Algorithm can be used to search documents with large data. This can be seen from the test results, namely the value of precision and recall close to value 1.
3) The Mixed Heuristic algorithm provides accurate results if the search query only uses 2 terms.
4) Ranking using the Mixed Heuristic algorithm gives good results, with an average value of 12% for incorrect results from the top 20 retrieved documents.

5. References

[1] A. A. Hassan, "Mixed heuristic algorithm for intelligent string matching for information retrieval," *Proc. - Sixth Int. Conf. Comput. Intell. Multimed. Appl. ICCIMA 2005*, pp. 11–16, (2005).

[2] B. Saini, V. Singh, and S. Kumar, "Information retrieval models and searching methodologies: Survey," *Int. J. Adv. Found. Res. Sci. Eng.*, vol. 1, no. 2, p. 20, (2014).

[3] M. M. Musthofa, "Implementation of Rabin Karp Algorithm for Essay Writing Test System on Organization xyz," *2019 Int. Conf. Inf. Commun. Technol.*, pp. 502–507, (2019).

[4] J. Savoy and E. Gaussier, "Information retrieval," *Agric. Meteorol.*, vol. 7, no. C, pp. 441–442, (2010).

[5] M. Adriani, J. Asian, B. Nazief, S. M. . Tahaghoghi, and H. E. Williams, "Stemming Indonesian: A Confi x-Stripping Approach," *ACMTransactions Asian Lang. Inf. Process.*, vol. 6, no. December 2007, pp. 307–314, (2007).

[6] N. P. Lestari, "*Uji Recall and Precision Sistem Temu Kembali*," Univ. Airlangga, (2016).