Integration Distance Similarity with Keyword Algorithm for Improving Cohesion between Sentences in Text Summarization

Rizki Darmawan¹ and Adi Wijaya²

¹ Computer Engineering Department, Sekolah Tinggi Teknik Muhammadiyah Cileungsi, Indonesia
² Computer Engineering Department, Mohammad Husni Thamrin University, Indonesia

Abstract. In recent time the exponential growth of textual information available on the Web, end user need to be able to access information in summary form. Commonly the method to get the summary is extraction method. One of extraction method that easier and commonly used is Keyword Algorithm, but this algorithm has a weakness in the cohesion between the sentences. Distance similarity method is one method used for solving the cohesion problem. The idea of this paper is to improve cohesion between sentences based on extraction of keyword algorithm. The hybrid keyword algorithm and the distance similarity method is proposed. The proposed method was compared three distance similarity such as Cosine, Dice and Jaccard that looking for the cohesiveness between sentences according to keyword algorithm extraction and performance as standard of evaluation. The result showed that Dice has the highest cohesion degree is 45.87 %. Although the best performance is Cosine that performance is influenced with gold standard of abstractive human summary.

1. Introduction

Recent research on who investigates a summary, how to make it and how to evaluate it is being intensively conducted. This can be happened because of an increase in it [1]. The summary becomes a very important thing because end users get the meaning of a reading that can be obtained quickly and does not destroy the existing meaning. The summary that get by extracting is obtained from a document after searching through the search engines [2]. The number of text-summarization research conducted by researchers proves that research in this field becomes a trend worthy of research.

The summary has the main purpose of presenting the main ideas in a document that create the document less. If a sentence has value as important as the other sentence will result in ineffective summaries in the absence of a reduction in the size of information. On the other hand the reduction of information will result in a proportional decrease in a document [3]. The information content in document appears in bursts, and one can therefore distinguish between more and less informative segments.

There are two ways for creating the summary: manually or automatically. Summaries that are done automatically are called auto summarization. For Creating auto summarization there are two methods that are extraction and abstraction. [4].

The method of making a summary extraction is a method that takes an important sentence from a reading while the abstraction method takes the essence of a reading text, information fusion, sentence compression and reformulation [5].

The method of extraction is easier to make than the abstraction method. The method of extraction is done by three steps that is making a representation between the original text, scoring sentence and choosing the highest sentence scoring. Some methods are used on extractives such as Naive-Bayes keyword extraction, Hidden Markov Model, Graph Method, Latent Sematic Indexing [4].
The important thing of Keyword extraction is mining the text of the document [6]. By extracting the appropriate keywords, we can easily select sentences in documents that have relationships with each other. The key extract calculation uses TF / IDF calculation, where this calculation indexes words in a sentence in a paragraph. The calculation is within the keyword extraction algorithm [7]. Research similar with these topics is called automatic term recognition in the context of computational linguistics, automatic indexing or automatic keyword extraction in research in information retrieval [8]. If someone wants to know the main statement of a paper, someone wants to have some keywords. The number of words is sometimes sufficient for the document overview; However, more powerful tools are desirable [9].

To represent a document, there are steps that are divided into paragraphs, sentences and tokens. Sometimes some preprocessing, such as stop word removal is also performed. The second step is to try to determine which sentence is important on a document [10]. Conversely abstractive summaries require machines to generate language and are difficult to replicate or extend to a wider domain.

Keyword Algorithm is easier and common in extracting summary. However, the algorithm has a problem in the lack of cohesion or correlation between sentences [11-12]. The correlation between sentences can be seen from how similar a sentence with other sentences in a text and to what extent the ideas in the text are expressed clearly by avoiding confusing information [11].

One way to resolve the problem of cohesion between sentences in extract summary is with determine the optimal combination between sentences [13]. The determination and cohesion optimization can be applied by using similarity method [14]. The function for similarity measure should be easy to compute, it should implicitly capture the relatedness of the documents, and it should also be explainable [15]. The similarity between two sentences, according to the vector representation described is calculated as the distance similarity [16].

The purpose of this research is to improve cohesion in the summary results based on keyword algorithm extraction combine with distance similarity method. Finally, our work of this paper is summarized in the last section.

2. Material and Method

The keyword algorithm uses the calculation of TF / IDF, which calculates the weight of a term that will determine performance a term in a document is in the corpus. Weighting is done by calculating a positive value for several times that the term occurs in a particular document, as well as a negative term to the number of documents containing the term. Consider term t and document d, where t appears in n of N documents in D. The TF-IDF function is of the form as follows:

\[ TFIDF (t,d,n,N)= TF (t,d) x IDF (n,N) \]

When the TF-IDF function is run against all terms in all documents in the document corpus, the words can be ranked by their scores. A higher TF-IDF score indicates that a word is both important to the document, as well as relatively uncommon across the document corpus. This is often interpreted to mean that the word is significant to the document, and could be used to accurately summarize the document. TF-IDF provides a good heuristic for determining likely candidate keywords, and it has been shown to be effective after several decades of research.

Similarity measured such as Jaccard \( sim (a,b) = |A \cap B| / |A \cup B| \), Dice \( sim (a,b) = 2.|A \cap B| / |A|+|B| \) and cosine coefficient \( sim (a,b) = |A \cap B| / \sqrt{|A|.|B|} \) are defining as arithmetic expression of cardinality of the set \( |A|, |B|, |A \cup B| \) and intersection \( |A \cap B| \) [17]. This coefficient among the most used similarity measures in science and are subject active research in many field such as approximate reasoning, decision making and fuzzy sets.

The summarization stage consists of three components i.e. the keyword extraction algorithm, the compression ratio selector and the cosine equation method. These three components will summarize the text being fed as a result of the final text summarized.
The first pre-processed document is tokenized by keyword extraction algorithm and then calculates TF/IDF for each term. Then sum all of TF/IDF term for each sentence and get sum of each sentence the next process is rank all of sentence based on sum of TF/IDF. After sentence is selected then perform calculation of their similarity with cosine, jaccard and dice method. After the calculation of the similarity method, the next process is re-arranging all of sentence based on the similarity from the highest to the lowest similarity. The compression ratio determines the position of sentence rank. In this study using a compression of 50% that means the sentence summary shrinkage 50% from the original text [18]. The new text with new sentence arrangement will be the final summarized text.

The resulting of extractive summary will be evaluated using two points of view, performance and degree of cohesion. F-Measure represents an evaluation for the performance of a summary while the degree of cohesion evaluates how closely the relationship of a sentence to another sentence. F-Measure is measuring how far the technique is capable of predicting of correct sentence. Evaluation can be classified into intrinsic and extrinsic evaluation [19]. Intrinsic evaluation judges the summary quality by its coverage between machine-generated summary and human generated summary. Extrinsic evaluation focuses mainly on the quality by its effect on other tasks. In intrinsic evaluation, Precision (P), recall (R), and F-measure (F) are used to judge the coverage between the manual and the machine generated summary:

\[
P = \frac{|S \cap T|}{|S|} \quad (1)
\]

\[
R = \frac{|S \cap T|}{|T|} \quad (2)
\]

\[
F = \frac{2 \times P \times R}{P + R} \quad (3)
\]

Where S is the machine generated summary and T is the manual summary [19]. For the cohesion evaluation, we can measure with the formula as follows:

\[
CoH = \frac{Log(C_s + 9 + 1)}{Log(M + 9 + 1)} \quad N_s = \frac{(o) \times (o - 1)}{2} \quad (4)
\]

\[
C_s = \frac{\sum_{S_i,S_j\in Summary} Sim_{cos}(S_i,S_j)}{N_s} \quad (5)
\]

\[
M = \max Sim_{cos}(i,j), \quad i,j \leq N
\]

\[
N_s = \frac{(o) \times (o - 1)}{2} \quad (6)
\]

Where CoH corresponds to the cohesion of a summary, Cs is the average similarity of all sentences in the summary S, Sim_{cos}(Si,Sj) is the cosine similarity between sentences Si and Sj, Ns is the number of nonzero similarity relationships in the summary, O is the number of sentences in the summary, M corresponds to the maximum similarity of the sentences in the document and N is the number of sentences in the document. In this way, CoH tends to zero when the summary sentences are too different among them, while that CoH tends to one when these sentences are too similar among them. Thus, this feature tends to favor the summaries that contain sentences about the same topic [12].
The dataset used in this research is collected from UCI Dataset containing documents of Reuters-21578 that has collection appeared on the Reuters newswire in 1987. The documents were assembled and indexed with categories by personnel from Reuters Ltd. (Sam Dobbins, Mike Topliss, and Steve Weinstein) and Carnegie Group, Inc. (Peggy Andersen, Monica Cellio, Phil Hayes, Laura Knecht, Irene Nirenburg) in 1987, https://archive.ics.uci.edu/ml/datasets/Reuters21578. In Figure 1 can be explained that after the data from UCI Reuters-21578 completed prepared then the data will be tested into summarization stage.

![Diagram](https://example.com/diagram.png)

**Fig. 1.** Block Diagram Proposed Model

### 3. Result and Discussion

The research using computer platform with specification based on Inter Core I5 2.70 Ghz 8 Gb RAM Microsoft Windows 7 Ultimate 64 Byte. The Software is using Java with Netbeans IDE 8.2. The evaluation of the result is using cohesion degree [12] that to know about how high related the sentences using combination of keyword extraction and distance similarity and using F-Measure to know performance of the summary as condition of intrinsic evaluation. In this experiment we use three distance similarity such as Cosine, Dice and Jaccard. This Similarity is very common used by researcher in many aspect [17].
After summarized using keyword algorithm with, the summary then re-calculated using the distance similarity for finding the best cohesion between sentences then compress half of the passage. The best cohesion then become candidate of sentence summary. Then we calculated the cohesion degree of the summary than can be seen in equation (4). Table 1 show the result of cohesion degree of summarization with distance similarity and without them

| Cohesion Degree from Distance Similarity (in percent) |
|------------------------------------------------------|
|            | Jaccard | Cosine | Dice  |
| cohesion degree | 41.50    | 44.30  | 45.87 |

From Table 1 can be seen that Dice similarity is the best cohesion than another distance similarity such as Jaccard and Cosine that has value 45.87%. The cohesion before re-calculated using distance similarity is 37.47%. The improve cohesion using Dice is 8.40 %, Cosine is 6.83% and Jaccard is 4.03%. Although in another experiment conclude that Cosine has best performance when be compared with Dice and Jaccard [20] but another reason perhaps Dice similarity measure cannot be induced when one vector is zero which overcomes the disadvantage of the cosine measure [21] and also its depend with document categorization from datasets [15].

Evaluation the performance of summarization using recall, precision and F-measure that the formula can be seen in equation 1 until 3 for three distance similarity show in Table 2 below.

| Recall, Precison and F-Measure |
|--------------------------------|
|            | Jaccard | Cosine | Dice  |
| Recall     | 39.67%  | 52.91% | 39.81% | 40.56% |
| Precision  | 35.42%  | 46.50% | 39.13% | 37.46% |
| F-Measure  | 37.12%  | 48.54% | 38.82% | 37.41% |

From Table 2 can be seen that Jaccard similarity such has Recall is 39.67 %, Precision 35.42 % and F-Measure 37.12 %. Cosine similarity has Recall 52.91 %, Precision 46.50 % and F-Measure is 48.54 %. Dice similarity has Recall 39.81 %, Precision 39.13 % and F-Measure 38.82 %. Based on this data, the best performance of the Distance similarity is Cosine similarity that has F-Measure is 48.54 % since the keyword itself has F-measure 37.40 %. So that the best improve performance from Cosine is 14.14 %. Although the best cohesion degree is Dice similarity but the best performance is Cosine. One reason to explain about this phenomenon is intersection human summary and machine summary is high. Intersection means how many words in machine summary have same similarity with number of word in human summary. If intersection is high, automatically make the high result although length of word in machine and human has big influence contribution to the result. This study has confirm that intersection between human summary and machine play big influence for evaluation measurement such as recall, precision and F-measure[11].

4. Conclusion

The main goal of summary is to present the main ideas in document in less space. If all sentences in a text document were equal importance, producing summary would not very effective, as any reduction in the size of a document would carry a proportional decrease in its informative.

In this research is used keyword algorithm model with three distance similarity. In this experiment, the best distance similarity that improved the cohesion degree is Dice similarity although the best performance is Cosine. Summarization with Cohesion degree not influence the performance because performance sometimes is measured compare with gold standard that be abstractive.
Acknowledgment

Thanks to Mr. Romi Satria Wahono that has guide us to make good research and encourage to present this research. Thanks to Sekolah Tinggi Teknik Muhammadiyah Cileungsi that support us to finish this paper.

Reference

[1] E. Hovy and C. Lin, “Automated and Text Summarization and The Summarist System,” Assoc. Comput. Linguist. (Proceeding TIPSTER ’98 Proc. a Work. held Balt. Maryland), pp. 197–214, 1998.

[2] A. Porselvi & S. Gunasundari, “Survey on web page visual summarization,” Int. J. Emerg. Technol. Adv. Eng., vol. 3, pp. 26–32, 2013.

[3] D. Radev, E. Hovy, and K. McKeown, “Introduction to the special issue on summarization,” Comput. Linguist., vol. 28, no. 4, pp. 399–408, 2002.

[4] D. Das and A. F. T. Martins, “A Survey on Automatic Text Summarization,” Eighth ACIS Int. Conf. Softw. Eng. Artif. Intell. Netw. ParallelDistributed Comput. SNPD 2007, vol. 4, pp. 574–578, 2007.

[5] R. M. Aliguliyev, “A new sentence similarity measure and sentence based extractive technique for automatic text summarization,” Expert Syst. Appl., vol. 36, no. 4, pp. 7764–7772, 2009.

[6] W. M. Wang, Z. Li, J. W. Wang, and Z. H. Zheng, “How far we can go with extractive text summarization? Heuristic methods to obtain near upper bounds,” Expert Syst. Appl., vol. 90, pp. 439–463, 2017.

[7] M. Rajman and R. Besan, “Text Mining - Knowledge extraction from unstructured textual data,” Proc. 6th Conf. Int. Fed. Classif. Soc., 1997.

[8] Y. MATSUO and M. ISHIZUKA, “Keyword Extraction From a Single Document Using Word Co-Occurrence Statistical Information,” Int. J. Artif. Intell. Tools, vol. 13, no. 01, pp. 157–169, 2004.

[9] H. P. Luhn, “A Statistical Approach to Mechanized Encoding and Searching of Literary Information,” IBM J. Res. Dev., vol. 1, no. 4, pp. 309–317, 1957.

[10] R. Ferreira, L. De Souza Cabral, R. D. Lins, G. Pereira E Silva, F. Freitas, G. D. C. Cavalcanti, R. Lima, S. J. Simske, and L. Favaro, “Assessing sentence scoring techniques for extractive text summarization,” Expert Syst. Appl., vol. 40, no. 14, pp. 5755–5764, 2013.

[11] K. Nandhini and S. R. Balasundaram, “Improving readability through extractive summarization for learners with reading difficulties,” Egypt. Informatics J., vol. 14, no. 3, pp. 195–204, 2013.

[12] M. Mendoza, S. Bonilla, C. Noguera, C. Cobos, and E. León, “Extractive single-document summarization based on genetic operators and guided local search,” Expert Syst. Appl., vol. 41, no. 9, pp. 4158–4169, 2014.

[13] M. A. Fattah and F. Ren, “GA, MR, FFNN, PNN and GMM based models for automatic text summarization,” Comput. Speech Lang., vol. 23, no. 1, pp. 126–144, 2009.

[14] Y. Bestgen, “Improving Text Segmentation Using Latent Semantic Analysis: A Reanalysis of Choi, Wiemer-Hastings, and Moore,” Comput. Linguist., vol. 32, no. 3, pp. 455–455, 2006.

[15] M. Rafi and M. S. Shaikh, “An improved semantic similarity measure for document clustering based on topic maps,” Comput. Sci. Dep. Karachi Pakistan, no. March, 2010.
[16] I. C. Mogotsi, “Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze: Introduction to information retrieval,” *Inf. Retr. Boston.*, vol. 13, no. 2, pp. 192–195, 2010.

[17] S. Jimenez, F. A. Gonzalez, and A. Gelbukh, “Mathematical properties of soft cardinality: Enhancing Jaccard, Dice and cosine similarity measures with element-wise distance,” *Inf. Sci. (Ny).* vol. 367–368, pp. 373–389, 2016.

[18] R. Darmawan and R. Wahono, “Hybrid Keyword Extraction Algorithm and Cosine Similarity for Improving Sentences Cohesion in Text Summarization,” *J. Intell. Syst.*, vol. 1, no. 2, pp. 109–114, 2015.

[19] K. Nandhini and S. R. Balasundaram, “Extracting easy to understand summary using differential evolution algorithm,” *Swarm Evol. Comput.*, vol. 16, pp. 19–27, 2014.

[20] V. Thada and V. Jaglan, “Comparison of Jaccard, Dice, Cosine Similarity Coefficient To Find Best Fitness Value for Web Retrieved Documents Using Genetic Algorithm,” *Int. J. Innov. Eng. Technol.*, vol. 2, no. 4, pp. 202–205, 2013.

[21] J. Ye, “Multicriteria decision-making method using the Dice similarity measure based on the reduct intuitionistic fuzzy sets of interval-valued intuitionistic fuzzy sets,” *Appl. Math. Model.*, vol. 36, no. 9, pp. 4466–4472, 2012.