Frequent Term based Text Summarization for Bahasa Indonesia

M.Fachrurrozi, Novi Yusliani, and Rizky Utami Yoanita

Abstract— Text summary helps in understanding the content of a text without having to read the contents of the text as a whole. Automatic text summarization can be used to summarize the text easier. In this paper a frequent term based text summarization for Bahasa Indonesia is designed and implemented in java. The proposed system generates a summary for a given input document based on identification and extraction of important sentences in the document. The system counts nouns and verbs term frequency because they are considered as the most representative to the content of the text. The system also integrated to statistical approach with two underlying concepts such as title of the news article and location of the sentence. The generated summaries were compared with human generated summaries. Precision, recall and f-measure ratio are used to evaluate the accuracy of the generated summary. Assessment of the system summary result quality by respondents is also done by giving a value from 1 to 100. Based on the experimental results, the system is able to produce an effective summary with the average f-measure of 78%, at the compression rate of 30%. The average value of the quality of system summary result provided by respondents is 83,3.

Keywords--- Automatic Text Summarization, Term Frequency,

I. Introduction

INFORMATION of a text can be gotten by read the entire content of the text, but it will require a longer time than read the summary of the text. With the summary information of a text can be obtained without having to read the entire text. This can avoid the reading of the text that is not relevant to the expected information, thus saving time used. Text summarization can be done by taking the core information from a text. Manual text summarization would require substantial time and cost to get a summary of the large amounts of text with the long contents. Automatic text summarization can be used to overcome this problem. Automatic text summarization is the process in which a computer creates a shorter version of the original text (or a collection of texts) still preserving most of the information present in the original text [7].

Automatic text summarization can be classified into two categories: extraction and abstraction [5]. Extraction technique summarize the text by copies the information that is considered the most important of the original text to be the summary, such as main clause, main sentence, or main paragraph. Abstraction technique transformed sentences to be the new shorter sentence that doesn't exist in the original text [4].

The proposed system applies frequent term based text summarization technique for text in Bahasa Indonesia Indonesian language. This study also uses the weighting calculation for each sentence based on the method of statistical approach text summarization taken by Y. Y. Chen and LH Chong's (2009) [3].

The rest of this paper is organized as follows. Section II presents related work. Section III describes preprocessing and proposed system, followed by experimental result in section IV. Finally, conclusions of the work in Section V**.**

II. Related Work

Manne and Suneetha propose a text summarization system by generates a summary for a given input document based on identification and important sentences in the document [6]. The system use frequent term based test summarization technique with HMM tagger. The summary is obtained by the ranked sentences that have been collected by identifying the feature terms.

Chong and Chen developed a text summarization for oil and gas news article [3]. Chong and Chen text summarization system is integrated statistical approach with three underlying concepts such as keyword occurences, title of the news article and location of the sentences.

In this paper, we proposed to incorporate the frequent term based text summarization technique with the statistical approach using the two underlying concepts such as title of the news article and location of the sentence to summarize the text.

III. Proposed System

The following Fig.1 represents the proposed system block diagram. The model consists of the following stages.

First of all text input document segmented into paragraphs. Paragraph segmentation is done by considering "\ n" as the marker of the end of the paragraph. After the input text split into paragraphs, sentence segmentation performed by considering full stop "." question mark "?", and the exclamation mark "!" as the marker of the end of the sentence.

M.Fachrurrozi is Lecturer in Faculty of Computer Science, Sriwijaya University, Indonesia.

Novi Yusliani is Lecturer in Faculty of Computer Science, Sriwijaya University, Indonesia

Rizky Utami Yoanita is Lecturer in Faculty of Computer Science, Sriwijaya University, Indonesia

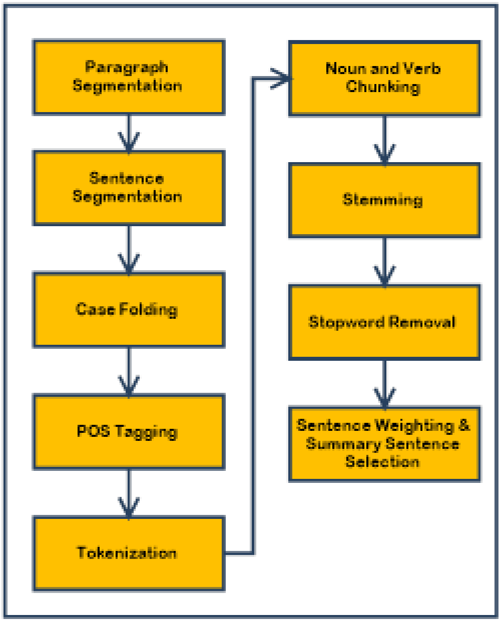

Fig.1 Proposed System Architecture

Once the text is segmented into sentences the process followed by case folding, the process to convert all the words in the sentence to lowercase and remove all punctuation there. This improves the accuracy of the system to distinguish similar words.

The next stage is POS tagging. POS tagging is the process of assigning Parts of Speech like noun, verb, pronoun, and so on to each word in a sentence to give word class. POS tagger which is used in this work is Hidden Markov Model based POS tagger for Bahasa Indonesia developed by Wicaksono and Purwanti, 2010 [8]. A POS tagger with HMM method was proved to have better running time than any other probabilistic methods. Both first order (bigram) and second order (trigram) Hidden Markov Model are used for developing the tagger system. Second order (trigram) Hidden Markov Model is used to find the best POS tag sequence on this work. Equation 1 is used to find the best POS tag sequence in second order HMM (trigram) case. Other improvement methods are used in HMM POS tagger for Bahasa Indonesia such as using affix tree to predict emission probability vector for OOV words and utilizing information from dictionary lexicon (*KBBI-Kateglo*) and succeeding POS tag.

$$t_{1-n} = \arg max_{t_{1...}t_n} P(t_1)\, x\; P(t_2|t_1)\; x \prod_{i=3}^{n} P(t_i|t_{i-1}, t_{i-2})\; x \prod_{i=1}^{n} P(w_i|t_i) \qquad (1)$$

The next stage is tokenization, splitting of the sentences into words by tracing white space as the separator. The process followed by noun and verb chunking to indentify noun and verb that is contained in the output of the previous result sentence. Noun and verb tag are NN (common noun), NNP (proper noun), NNG (genitive noun), VBI (intransitive verb), and VBT (transitive verb). Nouns and verbs derived from the noun and verb chunking process becomes the input for the stemming process.

Stemming is the process to get the root of the certain word in the document. By using the root, calculating the frequency of the word in the text will be more accurate. Stemming algorithm used is Indonesian stemming algorithms Nazief & Adriani. Indonesian stemming algorithms Nazief & Adriani has better percentage accuracy (precision) than any other algorithms [1].

After the stemming, process is continued by removing stopword. Stopwords are irrelevant words that in a text document. Stopword removal is done by comparing each word in the sentence with stoplist (stopword list) [2]. Examples of stopword are conjunctions, articles and prepositions. The output word from all of the process are called a term.

The next stage is Sentence weighting and summary sentence selection. Sentence weighting and summary sentence selection stage are the stage of determining sentence for the summary results by calculating the weight of the sentence and choose the sentence with the highest weight value. Weighting calculation begins by calculating the weight value of each term. Each term will be assigned with a basic weight (*wi*) using equation 2.

$$w_i = TF.ISF \qquad (2)$$

TF is the frequency of a particular word appears in the article and *ISF* are represented by equation 3.

$$ISF(w) = \log(S/SF(w)) \qquad (3)$$

w is a term, *S* is the number of sentences in a text document, *SF(w)* is the number of sentences that *w* occurred.

After assigning weight to each term, *wi* , all word's weight in a sentence are added up to represent the basic weight of the sentence *s*(*w*) shown in equation 4.

$$s(w) = \sum_{i=1}^{n} w_i \qquad (4)$$

s(*w*) is sum of word weight in a sentence, *n* is the number of words in the sentence, *wi* is the word weight of the *i*th word.

Terms of the sentence that are contained in the titles given extra weight as shown by equation 5, where *z* is number of term in the sentence that occur in the title.

$$St = 3z \qquad (5)$$

$$Sl = 5 \qquad (6)$$

$$Total\ Weight = St + Sl + s(w) \qquad (7)$$

In the calculation of the weighting, position of the sentence in the paragraph is also considered. The first sentence and the last sentence of a paragraph is given extra weight as shown in Equation 6. Finally, the total weight of the sentence is calculated based on the equation 7. The sentence that is used as summary sentences candidate only the third top and third bottom of the paragraph. Suppose the number of sentences in a paragraph is n, then 1/3n top and 1/3n bottom be the priority to generate a summary. After all the weighting and selection of candidate summary sentence is completed, each sentence has a total weight.

After all of the processes finish, the candidates of summary sentences are obtained. The candidates of summary sentences are sorted by the order of greatest weight to the smallest weight. The text documents were summarized at 30% compression rate. The summary sentences obtained then reassembled based on the original order of the text.

IV. Experimental Result

We compare 30 national news text document summarized by system and manually by human. The national news text documents were summarized at 30% compression rate by the system. The performance measures used to evaluate the quality of the summary is precision, recall and f-measure ratio which are shown accordingly in equation 8, 9 and 10 [9].

$$Precision = \frac{[\{Relevant\ sentences\} \cap \{Retrieved\ sentences\}]}{[\{Retrieved\ sentences\}]} \quad (8)$$

$$Recall = \frac{[\{Relevant\ sentences\} \cap \{Retrieved\ sentences\}]}{[\{Relevant\ sentences\}]} \quad (9)$$

$$f - measure = \frac{2 * precision * recall}{recall + precision} \quad (10)$$

Where *relevant sentences* are sentences that are identified in the human generated summary and *retrieved sentences* are sentences that are retrieved by the system. The highest *f-measure* value is 1 on 4 text document and the lowest value is 0.40 on 1 text document. The average value of *f-measure* is 0.78. We also conducted the questionnaire evaluation to 20 respondents. Questionnaire contains several questions for respondents' opinions about the system and the assessment of the system summary result quality. Assessment of the system summary result quality by the respondents is done by giving a value from 1 to 100. This value is given in accordance with the respondent's level of understanding of the system summary results and how well the results represent a summary of the contents of the system. The average system summary result quality value given by respondents is 83.3.

V. Conclusions

In this paper we have developed a frequent term based text summarization which is able to produce summary that can be understood by the respondents with an average value of 83.8 and has represented the content of the text as a whole. For future work, the system can be improved by integrating the system with another method to get the better summary. Besides being able to produce extraction the text summarization can also produces abstractions, the interpretation of the original text.

References

[1] Agusta, Ledy. *Perbandingan Algoritma Stemming Porter dengan Algoritma Nazief & Andriani untuk Stemming Dokumen Teks Bahasa Indonesia, Konferensi Nasional Sistem dan Informatika* (2009)

[2] Al-Hashemi, Rafeeq. *Text Summarization System (TSES) Using Extracted Keywords. International Arab Journal of e-Technology,* Vol.1, No.4 (2010)

[3] Chen , Yoke Yie and Chong, Ling Hui. *Text Summarization for Oil and Gas News Article,* Proceedings of World Academy of Science, Engineering and Technology 53 (2009)

[4] Gupta, Vishal and Lehal, Gurpreet Sigh. *A Survey of Text Summarization Extractive Techniques, Journal of Emerging in Web Intelligence* Vol.2 No.3 (2011)

[5] Mani, Inderjeet. *Summarization Evaluation: An Overview, The MITRE Corporation,* W640 11493 Sunset Hills Road Reston, VA 20190-5214 (2001)

[6] Suneetha, M and Fatima, S.Sameen. *Corpus based Automatic Text Summarization System with HMM Tagger,* International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1 Issue-3 (2011)

[7] Nagwani, Naresh Kumar and Verma, Dr. Shrish. *A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm,* International Journal of Computer Applications (0975-8887) Volume 17-No.2 (2011)

[8] Wicaksono, A. Farizki dan Purwanti, Ayu. *HMM Based Part-of-Speech Tagger for Bahasa Indonesia,* Proceeding of 4th International Malindo (Malay and Indonesian Language) Workshop (2010)

[9] Wu, Chia-Wei and Liu, Chao-Lin. *Ontology-based Text Summarization for Business News Articles.* Proceeding of the ISCA Eighteenth International Conference on Computer and Their Applications 389-392 (2003)