SINGLE DOCUMENT AUTOMATIC TEXT SUMMARIZATION USING TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)

Hans Christian[1]; Mikhael Pramodana Agus[2]; Derwin Suhartono[3]

[1,2,3]Computer Science Department, School of Computer Science, Bina Nusantara University,
Jln. K.H. Syahdan No 9, Jakarta Barat, DKI Jakarta, 11480, Indonesia
[1]hans.christian@binus.ac.id; [2]mikhael.oda@binus.ac.id; [3]dsuhartono@binus.edu

ABSTRACT

The increasing availability of online information has triggered an intensive research in the area of automatic text summarization within the Natural Language Processing (NLP). Text summarization reduces the text by removing the less useful information which helps the reader to find the required information quickly. There are many kinds of algorithms that can be used to summarize the text. One of them is TF-IDF (Term Frequency-Inverse Document Frequency). This research aimed to produce an automatic text summarizer implemented with TF-IDF algorithm and to compare it with other various online source of automatic text summarizer. To evaluate the summary produced from each summarizer, The F-Measure as the standard comparison value had been used. The result of this research produces 67% of accuracy with three data samples which are higher compared to the other online summarizers.

Keywords: *automatic text summarization, natural language processing, TF-IDF*

INTRODUCTION

In the recent years, information grows rapidly along with the development of social media. The information continues to spread on the internet especially in the form of the textual data type. For a short text data, it requires less amount of time for readers to know its contents. While, for a long text data, the entire text of the document should be reviewed to understand its contents, so it takes more effort and time. One possible solution from this problem is to read the summary. The summary is the simplified version of a document which can be done using a summarization tools. Summarization tools help the user to simplify the whole document and only showuseful information (Munot & Govilkar, 2014). However, to generate such summary is not that simple, it involves a deep understanding of the documents.

Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics. The “natural language” means a language that is used for daily communication by humans (Bird, Klein, & Loper, 2009). NLP is a computer science field to extract full meaning from data text. Linguistic concepts like part-of-speech (noun, verb, adjective, and others) and grammatical structure are commonly used in NLP. Aside from the part-of-speech and grammatical structure, NLP also has to deal with anaphora and ambiguities which often appear in a language. To deal with this, it requires knowledge representation, such as lexicon of words and their meanings, grammatical properties and a set of grammar rules, and sometimes other information such as thesaurus of synonyms or abbreviations (Kao & Poteet, 2007).

The research of summarization has been investigated by the NLP community for nearly the last half-century. Text summarization is the process of automatically creating a compressed version of a text that provides useful information for the users. A text that is produced from one or more texts conveys important information in the original text(s), and is no longer than half of the original text(s) and is significantly less than that (Radev et al., 2002). There are three important aspects that characterize research on automatic summarization from the previous definition. First, the summary may be produced from a single document or multiple documents. Second, the summary should preserve important information. Last, the summary should be short. In addition, Lahari, Kumar, and Prasad (2014) stated that sentences containing proper nouns and pronouns have greater chances to be included in the summary. These chancesare overcome through statistical and linguistic approach.

In general, there are two basic methods of summarization. They are extraction and abstraction. Abstractive text summarization method generates a sentence from a semantic representation and then uses natural language generation technique to create a summary that is closer to what a human might generate. There are summaries containing word sequences that are not present in the original (Steinberger & Ježek, 2008). It consists of understanding the original text and re-telling it in fewer words. It uses the linguistic approach such as lexical chain, word net, graph theory, and clustering to understand the original text and generate the summary. On the other hand, Extractive text summarization works by selecting a subset of existing words, phrases or sentences from the original text to form summary. Moreover, it is mainly concerned with what the summary content should be. It usually relies on the extraction of sentences (Das & Martins, 2007). This type of summarization uses the statistical approach like title method, location method, Term Frequency-Inverse Document Frequency (TF-IDF) method, and word method for selecting important sentences or keyword from document (Munot & Govilkar, 2014).

The Term Frequency-Inverse Document Frequency (TF-IDF) is a numerical statistic which reflects how important a word is to a document in the collection or corpus (Salton *et al.*, 1988). This method is often used as a weighting factor in information retrieval and text mining. TF-IDF is used majorly to stop filtering words in text summarization and categorization application. By convention, the TF-IDF value increases proportionally to the number of times that a word appears in a document, but is offset by the frequency of the word in the corpus, which helps to control the fact that some words are more common than others. The frequency term means the raw frequency of a term in a document. Moreover, the term regarding inverse document frequency is a measure of whether the term is common or rare across all documents in which can be obtained by dividing the total number of documents by the number of documents containing the term (Munot & Govilkar, 2014).

In this experiment, an extractive text summarization with TF-IDF method is used to build the summary. Through this experiment, the process of how a summary is formed by using the TF-IDF method is explained. The program is provided with three different documents to be summarized and to calculate its accuracy. An analysis is performed to find outhow the program can reach a certain precision.

Kulkarni and Apte (2013) mentioned that the better approach for extractive summarization program consists of 4 main steps. There are preprocessing of text, features extraction of both words and sentences, sentence selection and assembly, and summary generation. Figure 1 illustrates the procedures of a working extractive automatic text summarization program.

These steps have its respective tasks. First, preprocessing consists of the operation needed to enhance feature extraction such as tokenization, part of speech tagging, removing stop words, and word stemming. Second, it is feature extraction. It is used to extract the features of the document by obtaining the sentence in text document based on its importance and given the value between zero and one. Third, sentence selection and assembly are when the sentences are stored in descending order of the rank, and the highest rank is considered as the summary. Last, summary generation is the sentences that are put into the summary in the order of the position in the original document.

An example of these steps can be seen on the text summarization extraction system using extracted keywords program, as described by Al-Hashemi (2010). It accepts an input of a document. Then, the document is preprocessed by the system to improve the accuracy of the program to distinguish similar words. This preprocessing text includes stop word removal, word tagging and stemming. Next, the system uses frequency term, inverse document frequency, and existence in the document title and font type to distinguish relevant word or phrase. As the features are determined, the program starts to find the sentences with the given features and the additional characteristic such as sentence position in the document and paragraph, keyphrase existence, the existence of indicated words, sentence length, and sentence similarity to the document class. Figure 2 shows the diagram of the automatic text summarization extraction program to generate a summary.

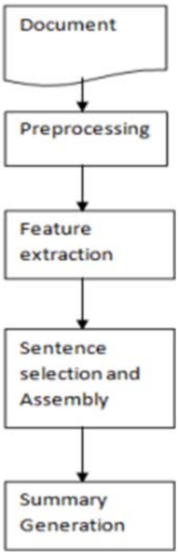

Figure 1 Steps of Extractive Automatic Text Summarization Process
(Source: Kulkarni & Apte, 2013)

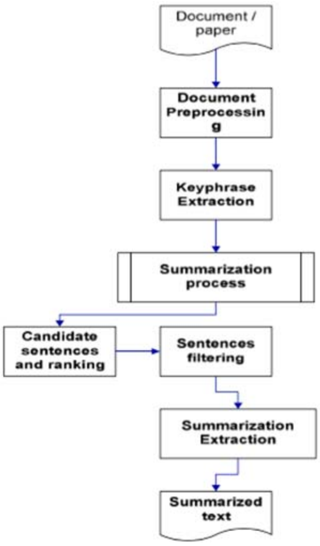

Figure 2 Diagram of the Text Summarization Extraction Program
(Source: Al-Hashemi, 2010)

METHODS

There is various kind of algorithm which can be used to create an automatic summarization. The most commonly used is an extractive text summarization with Term Frequency-Inverse Document (TF-IDF). This experiment aims to help the users to efficiently read the document(s) through summarization created by using this program. There are many existing tools which have the same automatic summarization function as this program, but the other programs only help to summarize the single document. This program is capable of summarizing multiple documents. However, in this experiment, the researchers only focus on the performance of the program in summarizing a single document. This experiment also calculates the accuracy of the summary produced by using TF-IDF compared to summary made by professional. The software architecture for this experiment can be seen in Figure 3.

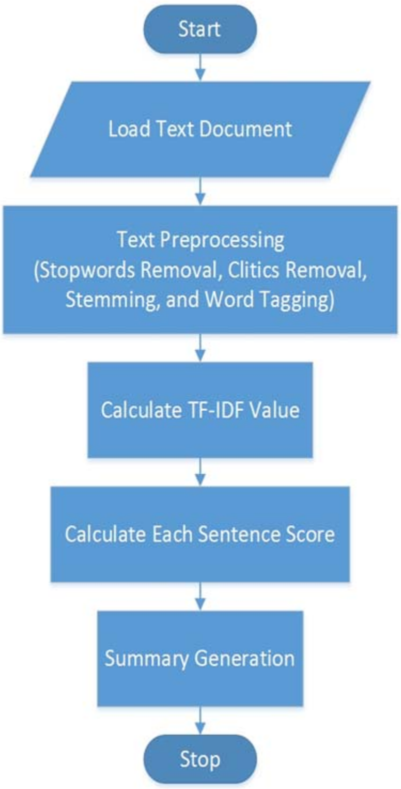
Start
Load Text Document
Text Preprocessing
(Stopwords Removal, Clitics Removal,
Stemming, and Word Tagging)
Calculate TF-IDF Value
Calculate Each Sentence Score
Summary Generation
Stop

Figure 3 Flowchart of Automatic Summarization

As mentioned before, TF-IDF is a numerical statistic which reflects on how important a word is to a document in the collection or corpus (Salton *et al*., 1988). The TF-IDF value increases proportionally to the number of times when a word appears in the document, but it is offset by the frequency of the word in the corpus, which helps to control the fact that some words are more common than others. The frequency term means the raw frequency of a term in a document. Moreover, the term regarding inverse document frequency is a measure of whether the term is common or rare across all documents in which can be obtained by dividing the total number of documents by the number of documents containing the term (Munot & Govilkar, 2014).

As this experiment is still in its early stage, the sample used in this experiment can only be pure text document copied into the program or ".txt" extension file. The sample is also used in calculating the accuracy of the summary. Three different documents are used as samples in this experiment. These three documents are descriptive text. Although the narrative text has also shown the readable result, in this experiment only descriptive text is used. Unfortunately, this experiment only calculates the result for single document summary as there is no comparison for multi-document summary. Despite not having any accuracy, the result for the multi-document summary can be deemed as readable. The maximum number of the document that the program can summarize are three documents in this stage.

Unlike the other artificial intelligence which needs machine learning, this automatic summarization experiment does not need any machine learning due to the use of existing libraries such as NLTK and TextBlob. By using these existing libraries, the experiment only focuses on how to calculate TF-IDF to summarize the text. This program is divided into three main functions which are preprocessing, feature extraction, and summarization.

Preprocessing function processes the document with NLTK functions like tokenization, stemming, part-of-speech (POS) tagger, and stopwords. After the document is inputted into the program, the preprocessing function splits the text into a list of words using tokenization functions. These tokenization functions are divided into two which are sentence tokenization and word tokenization. Sentence tokenization is a function to split the paragraph into sentences. While word tokenization is a function to split the string of written language into words and punctuation.

At first, the document is normalized using a lower function to normalize the text to lowercase so that the distinction between *News* and *news* is ignored. Then the paragraphs are tokenized into individual sentences. After that, the sentences are tokenized into a list of words. To make sure no unnecessary word in the list, every word in the list are classified using POS tagger function. This POS tag classifies the words into VERB (verbs), NOUN (nouns), PRON (pronouns), ADJ (adjectives), ADV (adverbs), ADP (adpositions), CONJ (conjunctions), DET (determiners), NUM (cardinal numbers), PRT (particles or other function words), X (other: foreign words, typos, abbreviations), '.' (punctuation) (Petrov, Das, & McDonald, 2012). Only VERB and NOUN are calculated in this experiment, because these types of the word are biased to make a summary (Yohei, 2002). All stopwords and clitics are also removed to prevent ambiguities. Then, the list of words is processed using stemming function to normalize the words by removing affixes to make sure that the result is the known word in the dictionary (Bird, Klein, & Loper, 2009).

From the preprocessed list of words, the TF-IDF value of each noun and verb can then be calculated. The equation of TF-IDF can be seen below.

$$TF = \frac{Totalappearanceofawordindocument}{Totalwordsinadocument} \tag{1}$$

$$\mathrm{IDF} = \log\frac{AllDocumentNumber}{DocumentFrequency} \tag{2}$$

$$TF - IDF = TF \times IDF \tag{3}$$

The value of TF-IDF ranges from zeroto one with ten-digit precision. After been calculated, these words are sorted in descending order by its value. Then, it is compiled into the new dictionary of word and its value. This sorting is important to analyze the rank of TF-IDF value from all of the words to check the output summary. After knowing TF-IDF value of each word, it can calculate the importance value of a sentence. The importance value of a sentence is a sum of the value of every noun and verb in the sentence. Every sentence in the document is sorted in descending order.

Finally, three to five sentences with the highest TF-IDF value are chosen. The number of sentences in the final summary may change depending on the compression rate of the program chosen by the user. As TF-IDF is an extraction method, the sentences that appear in the summary are the same as the original document. These chosen final sentences are sorted in accordance withits appearance in the original document. For the multi-document summarization, the sentences are sorted similarly with single document summarization. The difference is that it starts from the document which has the lowest total of TF-IDF.

RESULTS AND DISCUSSIONS

The program is created with Python Programming Languageand compiled in Microsoft Visual Studio 2015. Moreover, the interface of the program is created by using the Tkinter which is a package of Python Graphical User Interface. An additional package like Natural Language Toolkit and

Textblob are used for the text processing. Upon execution, the program asks user regarding how many documents to be summarized are. After all the documents are loaded, the user can determine how long the summary is generated by changing the compression rate of 10% and 90%. Then preprocessing step such as stopwords removal, stemming, and word tagging occur one at a time. Next, the program finds the features from the whole documents by using the statistical approach of frequency-inverse document frequency and performing selection to the sentence containing the features. The output summary is printed out in the output section of the interface, along with the percentage and statistical analysis of the summary.

In this experiment, the program has executedsix times, with a different set of documents and compression rate on each execution. The same documentis also summarized by two other online summarizers called www.tools4noobs.com/summarize and textsummarization.net/text-summarizer to be compared with the same compression rate. The statistical details on each experiment are displayed in the Table 1-10, along with the precision, recall, and f-measure of the program.

Table 1 Statistical Result of the Experiment

		Number of Sentences				
				Summary-Created by Summarizer		
Document	**Compression Rate**	**Original Document**	**Summary by Human**	**Program**	**Online summarizer 1**	**Online summarizer 2**
					(Tools 4 noobs)	**(Text Summarization)**
1	50%	14	7	7	8	7
2	70%	64	17	17	17	18
3	70%	18	5	5	5	5
4	80%	50	9	9	9	10
5	30%	28	19	19	19	19
6	50%	28	14	14	15	14

As shown in Table 1, during the first experiment of program the total number of the sentences in the original document is 14 and the summary by human consists of 7 sentences. With the compression rate is adjusted to 50%, program summarizer and online text summarizers 2 (Text Summarization) can produce 7 sentences while online text summarizer 1 (Tools 4 Noobs) produces 8 sentences for the summary. For the second experiment, the compression rate is increased to 70%, and the sentences in the third document are 64. Program summarizer and online summarizer 1 (Tools 4 Noobs) produce the same number of sentences as the summary by the human, which is 17, and online summarizer 2 (Text Summarization) has 18 sentences. In the third experiment, the document consists of 18 sentences and the summary by human consists of 5 sentences. This time the compression rate remains the same, and all the summarizers can produce the same number of the sentences with the summary by the human. During the fourth experiment, the document consists of 50 sentences and the compression rate is set to 80%. With that condition summary by human, program summarizer, and online summarizer 1 (Tools for Noobs) produce the same number of sentences which is 9. However, online summarizer 2 (Text Summarization) produces 10 sentences. The length of the fifth and sixth document is 28 sentences but with different compression rate which is 30% and 50%. The fifth experiment shows that all summaries produced have nine sentences. While the sixth experiment all summaries have 14 sentences, except the summary from online summarizer 1 (Tools 4 Noobs) which has 15 sentences.

Tabel 2 List of Top Words in the First Document Generated by Program Summarizer

Document 1	
Word	**TF-IDF value**
Caledonia	0,026322045
Island	0,01754803
Economy	0,01754803
Noumea	0,01754803
North	0,01754803

Tabel 3 List of Top Words in the Second Document Generated by Program Summarizer

Document 2	
Word	**TF-IDF value**
Holding	0,028407671
Door	0,024998751
Women	0,024998751
Men	0,023862444
Dating	0,015908296

Tabel 4 List of Top Words in the Third Document Generated by Program Summarizer

Document 3	
Word	**TF-IDF value**
Auctions	0,056815343
Object	0,034089206
Price	0,028407671
Participants	0,022726137
Example	0,017044603

Tabel5 List of Top Words in the Fourth Document Generated by Program Summarizer

Document 4	
Word	**TF-IDF value**
Music	0,0515891986
People	0,0171963995
Recorded	0,0132279996
Recording	0,0092595997
Played	0,0079367998

Tabel 6 List of Top Words in the Fifth Document Generated by Program Summarizer

Document 5	
Word	**TF-IDF value**
Internet	0,0364814306
Books	0,019456763
People	0,0145925722
Services	0,0145925722
Information	0,0121604769

Tabel 7 List of Top Words in the Sixth Document Generated by Program Summarizer

Document 6	
Word	**TF-IDF value**
Coffee	0,0712137514
Century	0,0118689586
Berries	0,0094951669
Drink	0,0094951669
Coffee-houses	0,0071213751

Table 2 to 7 describes the five most important words in each document. These words are selected based on the highest term frequency-inverse document frequency that is calculated after preprocessing step. In Table 2, the word “Caledonia” has the highest TF-IDF value among the other words. Hence the sentence consisting of this keyword generates higher sentence score than the other and most likely to be selected as a part of the summary.

Tabel 8 Program Summarizer Evaluation

Document	**Program Summarizer**					
	Correct	**Wrong**	**Missed**	**Precision**	**Recall**	**F-Measure**
1	5	2	2	0,714	0,714	0,714
2	10	7	7	0,588	0,588	0,588
3	4	1	1	0,800	0,800	0,800
4	7	2	2	0,778	0,778	0,778
5	11	9	8	0,550	0,579	0,564
6	8	7	6	0,533	0,571	0,552
Average				0,661	0,672	0,666

Tabel 9 Online Summarizer 1 (Tools 4 noobs) Evaluation

Document	**Online summarizer 1 (Tools 4 noobs)**					
	Correct	**Wrong**	**Missed**	**Precision**	**Recall**	**F-Measure**
1	6	2	1	0,750	0,857	0,800
2	9	8	8	0,529	0,529	0,529
3	3	2	2	0,600	0,600	0,600
4	7	2	2	0,778	0,778	0,778
5	10	9	9	0,526	0,526	0,526
6	7	7	7	0,500	0,500	0,500
Average				0,614	0,632	0,622

Tabel 10 Online Summarizer 2 (Text Summarization) Evaluation

Document	**Online summarizer 2 (Text Summarization)**					
	Correct	**Wrong**	**Missed**	**Precision**	**Recall**	**F-Measure**
1	5	2	2	0,714	0,714	0,714
2	9	9	8	0,500	0,529	0,514
3	4	1	1	0,800	0,800	0,800
4	6	3	3	0,667	0,667	0,667
5	10	9	9	0,526	0,526	0,526
6	8	7	6	0,533	0,571	0,552
	Average			0,623	0,635	0,629

According to Nedunchelian et al. (2011), the evaluation process of text summarization is performed by using three parameters which are precision, recall, and f-measure. Table 8, Table 9, and Table 10 represent the performance evaluation of the three different summarizers by using those parameters. The correct column shows the number of sentences that are extracted by the system and human; the wrong column shows the number of sentences that extracted by the system, and the missed column shows the number of sentences that extracted by the human.

The precision describes a ratio between the total of the relevant information and information which can be relevant or irrelevant to the system. The formula to calculate the precision can be seen below.

$$Precision = \frac{Correct}{Correct+Wrong} \tag{4}$$

On the other hand, recall describes a ratio between the total of the relevant information given by the system and the total of the relevant information which occurs inside the collection of information. The formula to calculate recall is as following.

$$Recall = \frac{Correct}{Correct+Missed} \tag{5}$$

Next, f-measure is a relationship between recall and precision which represent the accuracy of the system. The formula to calculate f-measure is in below.

$$F - measure = \frac{2\, xPrecisionxRecall}{Precision+Recall} \tag{6}$$

During the first experiment, online summarizer 1 (Tools 4 Noobs) produces correct and less missed sentences compared to the other summarizers. Therefore, this online summarizer produces the highest f-measure about 0,8. In the fourth experiment, program summarizer and online summarizer 1 (Tools 4 Noobs) produce the same number of the correct sentence which is 7 sentences. Moreover, online summarizer 2 (Text Summarization) produces only 6 correct sentences. However, in the second, third, fifth, and sixth experiment, program summarizer produces greater f-measure than the other online summarizers. Thus, the result of the average f-measure value from the six experiments is that program summarizer has the highest average f-measure about 0,666, the second one is online summarizer 2 (Text Summarization) with 0,629, and last the online summarizer 1 (Tools 4 Noobs) with 0,622.

CONCLUSIONS

This research explains the use of the algorithm of TF-IDF in an automatic text summarization program. Through this experiment, it can be seen that the TF-IDF algorithm can be used as the effective method to produce an extractive summary. It generates the summary with 67% of accuracy, which is a better result of the summary than other online summarizers.

From the comparison result between program summarizer and two online summarizers by using the statistical approach, it can be concluded that the program produces the better summary. By using the extractive method, TF-IDF is proven as a powerful method to generate the value which determines how important a word inside the document is. The value helps the program to determine which sentence to be used in the part of the summary.

There are some improvements that can be applied to this program to produce a more accurate summary. First, it is by making the summary biased on the title of the document. A title is a sentence or word that describes the main event or what the article is. Therefore, a high value of TF-IDF can be given to the word that appears in the title so that the program can produce a better result of the summary. Second, it is by increasing the number of experiment with a various type of sample document to increase the accuracy to calculate precision, recall, and f-measure value. It is because the more documents are summarized, the more valid the result of the average f-measure value becomes. Third, it should involve more respondents to evaluate the system by determining the number of correct, wrong, or missed sentences within the summary. This process will increase the validity of the experiment because the decision whether the sentence is the part of the summary is determined among the respondents.

REFERENCES

Al-Hashemi, R. (2010), Text Summarization Extraction System (TSES) Using Extracted Keywords, *International Arab Journal of E-Technology*, *1*(4), 164- 168.

Bird, S., Klein, E., & Loper, E. (2009) *Natural language processing with Python*. United States: O'Reilly Media.

Das, D., & Martins, A. F. (2007). A survey on automatic text summarization. *Literature Survey for the Language and Statistics*, *3*(3), 1-12.

Kao, A., & Poteet, S. R. (2007). *Natural Language Processing and Text Mining*. United States: Springer Media.

Kulkarni, A. R., & Apte, S. S. (2013). A domain-specific automatic text summarization using Fuzzy Logic. *International Journal of Computer Engineering and Technology (IJCET)*, *4*(4), 449-461.

Lahari, E., Kumar, D. S., & Prasad, S. (2014). Automatic text summarization with Statistical and Linguistic Features using Successive Thresholds. In *IEEE International Conference on Advanced Communications, Control and Computing Technologies 2014*.

Munot, N., & Govilkar, S. S. (2014). Comparative study of text summarization methods. *International Journal of Computer Applications*, *102*(12), 33-37.

Nedunchelian, R., Muthucumarasamy, R., & Saranathan, E. (2011). Comparison of multi document summarization techniques. *International Journal of Computer Applications. 11*(3), 155-160.

Petrov, S., Das, D., & McDonald R. (2012). A universal part-of-speech tagset. *arXiv preprint arXiv:1104.2086*.

Radev, D. R., Hovy, E., & McKeown, K. (2002). Introduction to the special issue on summarization. *Computational Linguistics, 28*(4), 399–408.

Salton, G., & Buckley, C. (1988). Term-Weighting approaches in Automatic Text Retrieval. *Information Processing and Management, 24*(5), 513–523.

Steinberger, J., & Ježek, K. (2008). Automatic Text Summarization (The state of the art 2007 and new challenges). *Znalosti*, *30*(2), 1-12.

Yohei, S. (2002). Sentence extraction by TF/IDF and Position Weighting from newspaper articles. In *Proceedings of the Third NTCIR Workshop*.