Summarizing Texts Automatically by Graph based Version of K Nearest Neighbor

Duke Taeho Jo
President
Alpha AI Publication
Cheongju, South Korea
tjo018@naver.com

***Abstract*—This article proposes the modified KNN (K Nearest Neighbor) algorithm which receives a graph as its input data and is applied to the text summarization. The graph is more graphical for representing a word and the text summarization is able to be viewed into a binary classification where each paragraph is classified into summary or non-summary. In the proposed system, a text which is given as the input is partitioned into a list of paragraphs, each paragraph is classified by the proposed KNN version, and the paragraphs which are classified into summary are extracted ad the output. The proposed KNN version is empirically validated as the better approach in deciding whether each paragraph is essential or not in news articles and opinions. In this article, a paragraph is encoded into a weighted and undirected graph and it is represented into a list of edges.**

I. Introduction

Text summarization refers to the process of selecting essential parts in each text. Each text is partitioned into sentences or paragraphs by punctuation mark or carriage return, respectively and the task is viewed as the binary classification of partitions into the essence partition or remaining. Each sentence or paragraph is encoded into its own structured form and sample sentences or paragraphs which are labeled with the essence part or the remaining are prepared. By learning sample ones, we construct the classification capacity, classify novice sentences into one of the two categories, and present the sentences or paragraphs which are labeled with the essences as the summary. We need to distinguish the summarization by system from one by human, in that text summarization by human is the process of rewriting a text into its brief version.

Let us mention some points which become the motivations for doing this research. The problems such as huge dimensionality and sparse distribution are caused by encoding texts into numerical vectors in using the traditional machine learning algorithms as the approaches [4]. Graphs are used as the popular representations of knowledge or information in the name of ontology or word net [1][14]. In recent works, various types of algorithms which manipulate graphs are developed correspondingly [14]. Therefore, by the motivations, in this research, we encode texts into graphs and modify the machine learning algorithm into its graph based version.

Let us mention what we propose in this research as some agenda. Instead of numerical vectors, we encode texts into graphs each of which consists of vertices indicating words and edges indicating their semantic relations. We define the similarity measure between graphs which have different vertices and edges as that between texts. We modify the KNN (K Nearest Neighbors) into its graph based version using the similarity measure, and apply it to the text summarization which is mapped into the binary classification task. The graphs which indicate texts are undirected weighted graphs and are represented into adjacency matrices in the implementation level.

Let us consider some benefits which are expected from this research. From using the proposed KNN version, we expect the better text summarization performance than from using the traditional version. By encoding texts into more graphical forms, we expect more transparency where we are able to guess the text contents only by their representations. We expect also more compactness in encoding texts into graphs than in doing them into numerical vectors; it causes the more efficient text processing. Hence, the goal of this research is to implement the text summarization system which satisfying the benefits.

This article is organized into the five sections. In Section II, we survey the relevant previous works. In Section III, we describe in detail what we propose in this research. In Section IV, we validate empirically the proposed approach by comparing it with the traditional one. In Section V, we mention the general discussion on the empirical validations and remaining tasks for doing the further research.

II. Previous Works

This section is concerned with the previous works which are relevant to this research. In Section II-A, we explore the previous cases of applying the KNN algorithm to text mining tasks. In Section II-B, we survey the schemes of encoding texts or words into structured data. In Section II-C and II-D, we survey the previous works on the string vector based machine learning algorithms and neural networks, respectively. Therefore, in this section, we provide the history about this research, by surveying the relevant previous works.

A. Related Tasks

This section is concerned with the previous cases of using the modernized KNN algorithm for the text summarization and its related tasks. Even if the text categorization is not covered in this research, it is regarded as the pivot task. We mention the text summarization which is covered in this research as the task which is derived from the text categorization, and present the cases of applying the modernized KNN algorithm for the task. We mention also the word categorization as another related task, and survey the cases of applying the modernized KNN algorithm for it. This section is intended to explore the previous cases of applying the modernized KNN algorithm for the text summarization, the text categorization, and the word categorization.

Let us explore the previous cases of applying the KNN version which deals directly with graphs to the relevant task to the text categorization. In 2016, Jo initiated modifying the KNN algorithm into the graph based version as the approach to the word categorization [17]. In 2018, he started to observe its performance by comparing the modernized version with the traditional version in classifying words [20]. In 2018, he validated the better performance of the modernized version than the traditional version as the approach to the word categorization through the three text collections [21]. In the above literatures, we cover the previous cases of using the graph based version of the KNN algorithm for the word categorization.

Let us explore the previous cases of applying the graph based version of the KNN algorithm for the text categorization as the base task. The graph based version was initially mentioned as the approach to the text categorization by Jo in 2018 [22]. In 2019, the graph based KNN version was compared with the traditional version, and its better performance is initially discovered in a small text collection [32]. The better performance of the graph based KNN algorithm was confirmed in the text categorization, though the three real collections, but it is not published yet [36]. In the above literatures, we presented the use of the graph based KNN algorithm for the base task as the better approach.

Let us review the previous works where the proposed KNN version is applied to the text summarization. The graph based KNN version was initiated as the approach to the text summarization, in 2017 [23]. It was compared with the traditional version, and its better performance was observed in a toy experiment, in 2018 [33]. This research aimed to finalize the empirical validation of the better performance of the graph based KNN algorithm in the text summarization. In the above literatures, we mentioned the application of the proposed KNN version to the text summarization.

We surveyed the previous works on the application of the modified version of the KNN algorithm to the tasks which are relevant to this research. The text summarization which is covered in this research is the process of extracting essential paragraphs as the summary from a text. The KNN version which is adopted in this research as the approach to the text summarization processes graphs directly. In the above literatures, the KNN version was applied as the approach to the word categorization and the text categorization. The research about the graph based version of the KNN algorithm for the text summarization has progressed, and the goal of this research is to complete validating empirically its better performance in the text summarization.

B. Encoding Schemes

This section is concerned with the schemes of encoding texts into structured data. In this research, it is proposed that texts should be encoded into graphs for modifying the KNN algorithm as the approach to the text summarization. In surveying the previous works, we mention the three structured data, numerical vectors, tables, and string vectors, as the alternative structured data. The encoding schemes are used for modifying the KNN algorithm and the AHC algorithm as the approach to the text mining tasks in the previous works. This section is intended to survey the previous works on the schemes where texts are encoded into the three types of structured data.

Let us survey the cases of encoding texts or words into numerical vector, in using the modernized machine learning algorithms to text mining tasks. In 2018, texts were encoded into numerical vectors in using the modernized version of the AHC algorithm as the approach to the text clustering [24]. In 2018, words were encoded into numerical vectors, in using the modernized version of the KNN algorithm as the approach to the word categorization [25]. In 2019, texts were encoded so in using the KNN algorithm to the text categorization [34]. In the above literatures, the KNN algorithm and the AHC algorithm were modernized by considering the feature similarities and the feature value similarities, in computing the similarity between numerical vectors.

Let us survey the previous cases of encoding texts into tables. Texts were initially encoded into tables in the text categorization by Jo and Cho in 2008 [11]. Texts were encoded for modifying the online linear clustering algorithm as the approach to the text clustering [8]. The table matching algorithm was proposed as the better and more stable approach to the text categorization in 2015 [16]. In the above literatures, we presented the cases of encoding texts into tables, instead of numerical vectors.

Let us mention the previous works where texts are encoded into string vectors in applying the machine learning algorithms to the text mining tasks. In 2018, the KNN algorithm was modified into the string vector based version which processes string vectors directly as the approach to the text categorization [26]. In 2018, the KNN algorithm was applied to the text summarization which is mapped into the binary classification where each paragraph is classified into summary or non-summary [27]. In 2020, the AHC algorithm as well as the KNN algorithm was modified into the string vector based version as the approach to the text clustering [37]. In the above literatures, we presented the cases of encoding texts into string vectors.

We surveyed the previous works on how to encode texts into structured data. Although texts or words are encoded into numerical vectors as the traditional structured form, the similarity metric is defined, considering the feature similarities, in order to avoid the poor discriminations among sparse vectors. They are encoded into tables, and the similarity metric was defined based on shared words. They are encoded into string vectors, and the similarity metric was defined as a semantic operation on string vectors. In this research, texts are encoded into graphs, and each graph consists of words as its vertices and semantic relations among them as its edges.

C. String Vector based Machine Learning Algorithms

This section is concerned with the previous works on the string vector based machine learning algorithms, as a kind of non-numerical vector based ones. A string vectors, each of which is an ordered finite set of strings, become the structured data where numerical values are replaced by strings as their elements. In this section, as the typical string vector based machine learning algorithms, we mention the string vector based KNN algorithm, the string vector based AHC algorithm, and the SVM (Support Vector Machine) with the string vector kernel. The semantic similarity between strings was defined under the assumption of each string with its own meaning in the previous works. This section is intended to explore the previous works on the three string vector based machine learning algorithms.

Let us survey the previous works on the string vector based KNN algorithm as a classification algorithm where the input data is given as a string vector. It was initially proposed as the approach to the word categorization, in 2018 [28]. The version of the KNN algorithm was applied to the text categorization as well as the word categorization, in 2018 [29]. It was applied to the text summarization which is mapped into a binary classification of each paragraph into summary or non-summary [30]. In the above literatures, we presented the string vector based KNN algorithm, as a non-numerical vector based machine learning algorithm for avoiding the problems in encoding texts or words into numerical vectors.

Let us explore the previous works on the string vector based AHC algorithm which clusters string vectors, directly. In 2018, the AHC algorithm was modified into the version which processes string vectors directly by defining the similarity between string vectors, and applied to the word clustering [31]. In 2019, the modified AHC algorithm was applied to the text clustering, as well as the word clustering [35]. In 2020, its better performance than the traditional version was validated in the three text collections, completely [38]. In the above literatures, we present the AHC algorithm which clusters string vectors as a non-numerical vector based clustering algorithm.

Let us mention the previous works on the string vector kernel based learning algorithm which processes string vectors as another string vector based one. The string vector kernel function was defined as the similarity between string vectors which is computed by the inverted index of strings, in 2007 [6]. The similarity matrix where its columns, its rows, and its elements correspond respectively, to a string and a semantic similarity between strings, was defined as the basis for computing the similarity between two string vectors as the string vector kernel, in 2007 [7]. The string vector kernel which is proposed in the above literatures was applied for modifying the SVM (Support Vector Machine) as the approach to the text categorization, in 2008 [9]. In the above literatures, we mention the string vector kernel function which is defined as the similarity between string vectors, used for modifying the SVM.

We surveyed the previous works on the string vector based machine learning algorithms which process string vectors directly instead of numerical vectors. The raw data such as textual data was encoded into string vectors for using this kind of non-numerical vector based machine learning algorithms. In surveying the previous works, we mention the string vector based KNN algorithm as the approach to the classification tasks and the string vector based AHC algorithm as the approach to the clustering tasks. The similarity between string vectors was defined as the string vector kernel, and used for modifying the SVM. In this research, we propose the graph based KNN algorithm as the alternative kind of the non-numerical vector based machine learning algorithms.

D. String Vector based Neural Networks

This section is concerned with the previous works on the string vector based neural networks. The two neural networks, NTC (Neural Text Categorizer) and NTSO (Neural Text Self Organizer), which process string vectors directly, were invented. They were used as the approaches to the text categorization and the text clustering. The NTC was cited as one of innovative ones, in the previous works on the text categorization. This section is intended to survey the previous works on the two neural networks as string vector based neural networks.

Let us survey the previous works on the NTSO which processes string vectors directly, as an unsupervised neural networks. It was initially proposed by Jo and Japkowicz, in 2005 [3]. The NTSO was mentioned as an innovative neural networks, in that it processes string vectors directly instead of numerical vectors, by Zheng et al. in 2006 [5]. Its better clustering performance, compared with the online linear clustering algorithm and the k means algorithm was validated empirically, in 2010 [12]. In the above literatures, we present the NTSO as the innovative neural networks which were applied to the text clustering.

Let us survey the previous works on the NTC (Neural Text Categorizer) as a string vector based neural networks. It was initially created as the approach to the text categorization, by Jo, in 2000 [2]. It was improved by adding automatic weight updating process, in 2008 [10]. Its better performance than the KNN and the SVM as the main approaches was empirically validated in both the hard text categorization and the soft text categorization in 2010 [13]. In the above literatures, we present the initial creation, the improvement, and the validation of the NTC which is a text classification tool.

Let us survey the previous works which citing the NTC. It was invented by Jo, and used for classifying Arabian texts by Abainia et al. in 2015 [15]. It is mentioned as an innovative approach to the text categorization by Vega and Mendez-Vazquez in 2016 [18]. It is mentioned in proposing the application of neural networks to the web page classification with the PCA (Principal Component Analysis) by Flaih in 2017 [19]. In the above literatures, we present the application and the citation of the NTC.

We reviewed the previous works on the two string vector based neural networks: the NTC and the NTSO. The NTC which belongs to the supervised neural networks was proposed as the approach to the text categorization, whereas the NTSO which belongs to the unsupervised neural networks was proposed as the approach to the text clustering. The NTC may be applied to the text summarization by mapping it into the classification task. The NTSO which was initially proposed as the approach to the clustering may be converted into its supervised version as the approach to the classification task. In next research, we will consider the two string vector based neural networks to the text summarization.

III. Proposed Approach

This section is concerned with encoding words into graphs, modifying the KNN (K Nearest Neighbor) into the graph based version and applying it to the text summarization, and consists of the three sections. In section III-A, we deal with the process of encoding texts into graphs. In section III-B, we describe formally the process of computing the similarity between two graphs. In section III-C, we do the graph based KNN version as the approach to the text summarization. In section III-D, we present the system architecture and the execution flow of the proposed system.

A. *Text Encoding*

This section is concerned with the process of encoding a text into a graph. In the context of the data structure, the graph is defined as its vertex set and its edge set. In the graph which represents a text, the words are given as vertices and the similarities among words are given as edges. The semantic similarity between words is computed by their collocations among texts in the corpus. This section is intended to describe the process of mapping a text into a graph and presenting a graph representing a text as an example.

The process of mapping a text into N words which is given as vertices is illustrated in Figure 1. When encoding a text into a graph, the words are defined as the vertices. In Figure 1, a single text is given as the input in the left side, and the k words are given as the output in the right side. The k words are generated through the three basic steps: the tokenization, the stemming, and the stemming removal. The vertex set is built in this step by indexing the text into a list of words as shown in Figure 1.

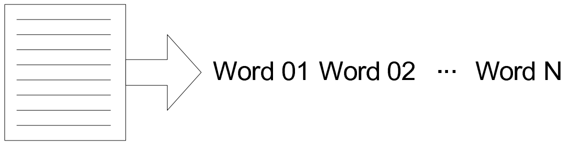

Figure 1. Text Indexing

The definition of the similarity matrix for generating edges in the graph which represents a text is illustrated in Figure 2. The N words are collected by indexing a text as shown in Figure 1. All possible pairs are generated from them, and the similarity matrix is constructed by computing the similarity for each pair by the equation which is presented in Figure 2. The similarity between two words is always given as a normalized value between zero and one. Some edges with their higher similarities than the threshold may be selected, and the threshold is set between zero and one, as an external parameter.

$$\begin{array}{cc} & \begin{array}{cccc}\text{Word 1} & \text{Word 2} & \text{....} & \text{Word N}\end{array} \\ \begin{array}{c}\text{Word 1}\\ \text{Word 2}\\ \text{....}\\ \text{Word N}\end{array} & \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1N} \\ s_{21} & s_{22} & \cdots & s_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ s_{N1} & s_{N2} & \cdots & s_{NN} \end{bmatrix} \end{array} \quad s_{ij} = \frac{2 \times \#\left(word_i, word_j\right)}{\#\left(word_i\right) + \#\left(word_j\right)}$$

Figure 2. Similarity Matrix

A simple example of the graph which represents a text is illustrated in Figure 3. The four vertices are defined and given as the four words: information, system, business, and computer. The complete links among the four words are six edges each of which is weighted. The weight which is associated with each edge is a similarity between two words. The graph which represents a text belongs to the undirected and weighted graph.

Let us make some remarks on the process of encoding texts into graphs. The graph is defined in the context of data structures as its vertex set and its edge set. In representing a

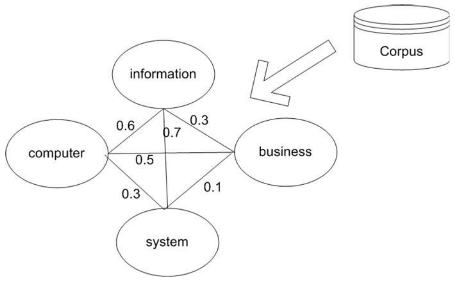

Figure 3. Graph representing a Text

text into a graph, words in the text are given as vertices, and the similarities among them are given as edges. In this research, a graph is viewed as an edge set in the implementation level. In future, we need to define more operations on graphs for modifying other machine learning algorithms.

B. Similarity between Two Graphs

This section is concerned with the process of computing the similarity between graphs. A graph is viewed as a set of edges in the implementation level. The similarity between edges is defined and the similarity between graphs is done afterward by means of one between an edge and a graph. The similarity between graphs is always given as a normalized value between zero and one, and proportional to the number of shared edges. This section is intended to describe the computation of similarity between graphs, in detail.

The three cases which are considered in computing a similarity between two edges is illustrated in Figure 4, and the two edges are defined as the entries, each of which consists of its two vertices and its weight, as shown in equation (1),

$$e_1 = (v_{11}, v_{12}, w_1), e_2 = (v_{21}, v_{22}, w_1) \tag{1}$$

If two vertices are same to each other in the two edges as shown in the left of Figure 4, the two edge weights are averaged as the similarity between edges, as shown in equation (2),

$$\begin{aligned}&\text{if } ((v_{11} = v_{21}) \wedge (v_{12} = v_{22})) \vee ((v_{11} = v_{22}) \wedge (v_{12} = v_{21}))\\ &\text{then } sim(e_1, e_2) = \frac{1}{2}(w_1 + w_2)\end{aligned} \tag{2}$$

If either of the two vertices is same to each other in two edges, as shown in the middle of Figure 4, the product of two weights is the similarity between edges, as shown in equation (3),

$$\begin{aligned}&\text{if } (((v_{11} = v_{21}) \wedge (v_{12} \neq v_{22})) \vee ((v_{11} = v_{22}) \wedge (v_{12} \neq v_{21}))\\ &\vee ((v_{11} \neq v_{21}) \wedge (v_{12} = v_{22})) \vee ((v_{11} \neq v_{22}) \wedge (v_{12} = v_{21})))\\ &\text{then } sim(e_1, e_2) = w_1 \cdot w_2\end{aligned} \tag{3}$$

If any vertex is not same to each other in the two edges as the right of Figure 4, the similarity between the edges becomes zero, as shown in equation (4),

$$\begin{aligned}&\text{if } ((v_{11} \neq v_{21}) \wedge (v_{12} \neq v_{22})) \vee ((v_{11} \neq v_{22}) \wedge (v_{12} \neq v_{21}))\\ &\text{then } sim(e_1, e_2) = 0\end{aligned} \tag{4}$$

In computing the similarity between the two edges, it is assumed that the weight which is assigned to each edge is always given as a normalized value between zero and one.

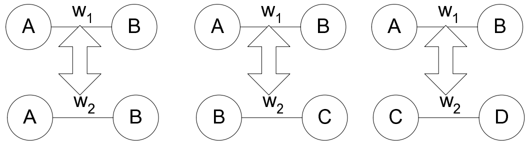

Figure 4. Three Cases in computing Edge Similarity

Let us compute the similarity between an edge and a graph by expanding one between edges. The similarity between two edges, $sim(e_1, e_2)$, is computed by the above process, and the similarity between an edge and a graph, $sim(e_1, G_2)$, where $G_2 = \{e_{21}, e_{22}, \ldots, e_{2|G_2|}\}$, is done, now. The maximum of the similarities of the edge, e_1, with the edges of the graph, G_2, is the similarity, $sim(e_1, G_2)$, as expressed by equation (5),

$$sim(e_1, G_2) = \max_{i=1}^{|G_2|} sim(e_1, e_{2i}) \tag{5}$$

$e_{\max}$ is the edge of the graph, G_2, which satisfy equation (6), as the most similar one as the edge, e_1

$$\max_{i=1}^{|G_2|} sim(e_1, e_{2i}) = sim(e_1, e_{\max}) \tag{6}$$

We need to remove the edges with no vertex which is shared by the edge, e_1, in the graph, G_2, in advance, for the more efficient computation.

Let us compute the similarity between two graphs by expanding one between an edge and a graph. The two graphs, G_1 and G_2 , are expressed respectively into the two sets, $G_1 = \{e_{11}, e_{12}, \ldots, e_{1|G_1|}\}$ and $G_2 = \{e_{21}, e_{22}, \ldots, e_{2|G_2|}\}$. The similarity between G_1 and G_2 is computed by equation (7),

$$sim(G_1, G_2) = \frac{1}{|G_1|} \sum_{i=1}^{|G_1|} sim(e_{1i}, G_2) \tag{7}$$

The similarity between two graphs is always a normalized value between zero and one, as shown in equation (8),

$$0 \leq sim(G_1, G_2) \leq 1 \tag{8}$$

The similarity metric which is expressed in equation (7), is used for modifying the KNN algorithm into the graph based as the approach to the text categorization.

Let us make some remarks on the similarity between graphs which was described in this section. The graph which represents a text is defined as its vertices as words and its edges as semantic relations among them. The three cases are considered in computing the similarity between two edges. The similarity between two edges is expanded into one between an edge and a graph, and it is expanded one more time into one between two graphs. The similarity metric between two graphs is utilized for modifying the KNN algorithm as the approach to the text summarization, in this research.

C. Proposed Version of KNN

This section is concerned with the graph based version of the KNN algorithm which process graphs directly. In the previous section, we described the similarity metric between two graphs, under the representation of each graph into an edge set. In the proposed KNN algorithm, a novice text is encoded into a graph, and its similarities with the training graphs is computed by the similarity metric. The text summarization is viewed into a binary classification of the paragraphs, and the proposed version of the KNN algorithm is adopted for implementing the text summarization system. This section is intended to describe the proposed version of the KNN algorithm which classifies a graph, directly.

Figure 5 illustrated that the similarities of a novice graph with the sample graphs are computed for selecting nearest neighbors. A novice text is encoded into the graph, G_{nov}, the predefined categories are notated by $C = \{c_1, c_2, \ldots, c_{|C|}\}$, and the training set which consists of n sample graphs which represent the sample texts is notated by $Tr = \{(G_1, y_1), (G_2, y_2), \ldots, (G_n, y_n)\}$, where G_i is a sample graph, and $y_i \in C$. The similarities of the novice graph, G_{nov} with the sample graphs, $G_1, G_2, \ldots, G_n$, are computed by equation (7), as $sim(G_{nov}, G_1), sim(G_{nov}, G_2), \ldots, sim(G_{nov}, G_n)$ in the proposed KNN algorithm. The similarity between the novice graph, G_{nov}, and a sample graph, is given as a normalized value between zero and one, as shown in equation (8). The similarities, $sim(G_{nov}, G_1), sim(G_{nov}, G_2), \ldots, sim(G_{nov}, G_n)$ are ranked by their values for selecting nearest neighbors.

The process of selecting nearest neighbors after computing their similarities with the novice item is illustrated in Figure 6. The similarities which are computed by equation (7) are ranked into ones, $sim(G_{nov}, G'_1), sim(G_{nov}, G'_2), \ldots, sim(G_{nov}, G'_n)$. The K items with their highest similarities with the novice item are selected as its nearest neighbors, as expressed in equation (9),

$$Near(K, G_{nov}) = \{G'_1, G'_2, \ldots, G'_K\} K \ll N \tag{9}$$

As an alternative way, we may consider selecting items with their higher similarities than a given threshold. We use the nearest neighbors,$G'_1, G'_2, \ldots, G'_K$ from the training examples, for deciding the label of the novice graph, G_{nov}.

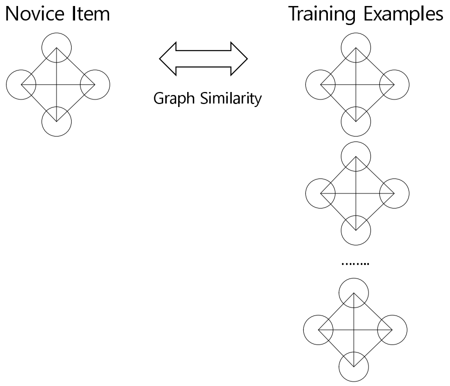

Figure 5. Similarities of a Novice Graph with Sample Ones

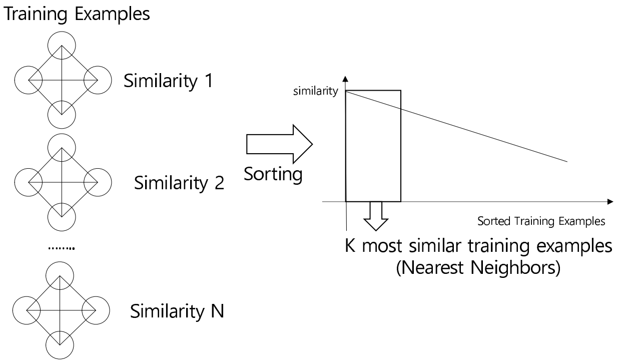

Figure 6. Selection of Nearest Neighbors from Training Examples

The process of voting the labels of the nearest neighbors for deciding the label of the novice item is illustrated in Figure 7. The nearest neighbors are selected by the process which is illustrated in Figure 7, as a set, $Ne = \{G'_1, G'_2, \ldots, G'_K\}$, and the function for weighting a nearest neighbor by a category is defined as equation (10),

$$w(C_i, G'_j) = \begin{cases} 1 & \text{if } G'_j \in C_i \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

For each category, the number of nearest neighbors which belong it is counted as shown in equation (11),

$$Count(C_i, Ne) = \sum_{j=1}^{K} w(C_i, G'_j) \tag{11}$$

The label of a novice item is decided by the label with the majority of the nearest neighbors, $C_{\max}$, as shown in equation (12),

$$C_{\max} = \operatorname*{argmax}_{i=1}^{|C|} Count(C_i, Ne) \tag{12}$$

The function, $w(C_i, G'_j)$ may be expanded into $w(C_i, G'_j, G_{nov})$ by augmenting the novice item, if the weight is dependent on the distance between the nearest neighbor and the novice item.

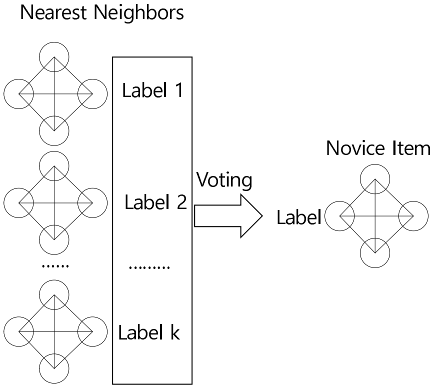

Figure 7. Voting Labels of Training Examples for deciding One of Novice Example

Let us make some remarks on the graph based version of the KNN algorithm which is proposed as the approach to the text summarization. Texts are encoded into graphs for using the version of the KNN algorithm, instead of numerical vectors. The similarity metric between graphs which was described in Section III-B is used for computing the similarities of a novice text with the sample texts. The graphs which represent the sample texts are ranked by their similarities with the notice one, and the K samples are selected as the nearest neighbors. The labels of the nearest neighbors are voted for deciding the label of the novice one.

D. Text Summarization System

This section is concerned with the system architecture and the execution process of the text summarization system. The text summarization system is mapped into the binary classification of a paragraph into summary or non-summary, and the KNN algorithm which was described in Section III-C, is adopted for implementing the system. The text which is given as the input is partitioned into paragraphs, and they are classified into summary or non-summary, and ones which are classified into summary are extracted as the output, in the system. We present the system architecture and the execution process in the design step, but omit the source code which implements the system in Java or Python. This section is intended to describe ones in the design level about the system.

The process of sampling paragraphs and classifying a novice one is illustrated in Figure 8. Because even a same paragraph may be classified differently depending on its domain, the sample paragraphs which are labeled with summary or non-summary are gathered domain by domain. A text which is tagged with its domain is given as the input, and the paragraphs in the text are classified into summary or non-summary by the classifier which corresponds to the domain. It requires to tag the text for performing the text summarization. Automating tagging the text with its domain will be considered in the next research.

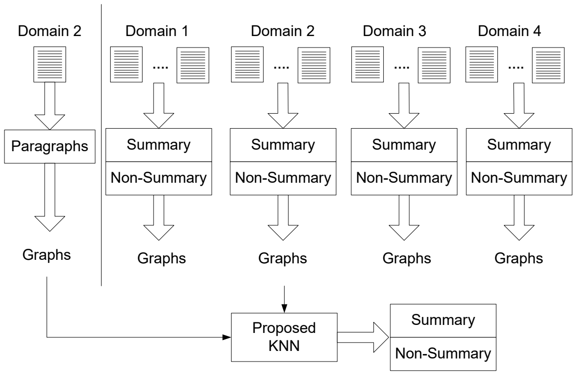

Figure 8. Process of Collecting Sample Paragraphs

The system architecture of the text summarization system is illustrated in Figure 9. The text partition module partitions an input text into paragraphs, and the encoder module encodes them into graphs by the process which was described in Section III-A. The similarity computation module computes the similarities of each paragraph with the sample paragraphs which are labeled with summary or non-summary by the process which was described in Section III-B, and selects some samples with their highest similarities as the nearest neighbors as the core part of the system. The paragraphs which are generated from the input text are classified into summary or non-summary, and ones which are classified with summary are extracted as the final output.

The execution flow of the text summarization system is illustrated in Figure 10. Texts are initially collected within a domain, their paragraphs are extracted from them, and they are labeled with summary or non-summary, manually. The labeled paragraphs and the novice paragraphs which are extracted from an input text are encoded into graphs.

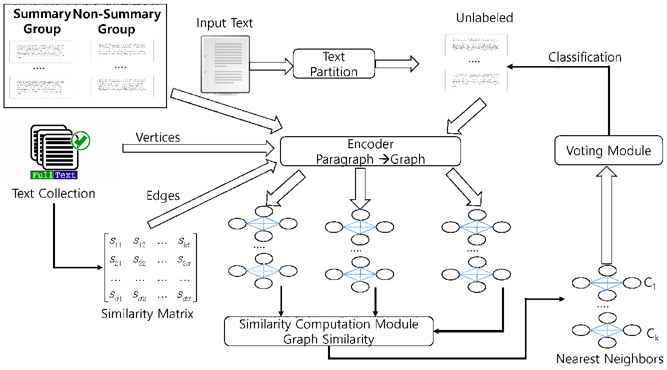

Figure 9. System Architecture

The novice paragraphs are classified by the KNN algorithm which is described in Section III-C, and there are two groups of paragraphs in the text: the summary group and the non-summary group. In this system, the paragraphs in the summary group are extracted as the summary of the input text.

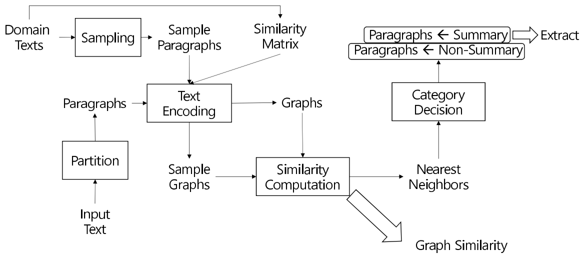

Figure 10. Execution Process

Let us make some remarks on the system architecture and the execution process of the text summarization system which are presented in Figure 9 and 10. In this research, the text summarization is mapped into a binary classification of paragraphs into summary or non-summary, and it is proposed that they are encoded into graphs. The KNN algorithm is modified into its graph based version as the approach to the text summarization, using the similarity between graphs. In this research, we present the system architecture and the execution flow which are necessary for doing the general design of the system. The real implementation of the text summarization system with the Java and the Python will be considered in the next research; we present the diagrams and the source codes for implementing the system in the next research.

IV. Experiments

This section is concerned with the empirical experiments for validating the proposed version of KNN, and consists of the five sections. In Section IV-A, we present the results from applying the proposed version of KNN to the text summarization on the collection, NewsPage.com. In Section IV-B, we show the results from applying it for classifying paragraphs into summary or not, from the collection, Opinosis. In Section IV-C and IV-D, we mention the results from comparing the two versions of KNN with each other in the task of text summarization from 20NewsGroups.

A. *NewsPage.com*

This section is concerned with the experiments for validating the better performance of the proposed version on the collection: NewsPage.com. We interpret the text summarization into the binary classification where each paragraph is classified into summary or non-summary, and gather the paragraphs which are labeled with one of the two categories, from the collection, topic by topic. Each paragraph is classified exclusively into one of the two labels. We fix the input size as 50 dimensions of numerical vectors, and use the accuracy as the evaluation measure. Therefore, this section is intended to observe the performance of the both versions of KNN in the four different domains.

In Table I, we specify the text collection, NewsPage.com, which is used in this set of experiments. The collection was used for evaluating approaches to text categorization tasks in previous works [16]. In each category, we extract 250 paragraphs and label them with summary or non-summary, keeping the complete balance over the two labels. In each category, the set of 250 paragraphs is partitioned into the training set of 200 paragraphs and the test set of 50 ones. Each text is segmented into paragraphs by a carriage return, and they are corrected manually, in the process of extracting paragraphs.

Table I
The Number of Texts and Paragraphs in NewsPage.com

Category	#Texts	#Training Paragraphs	#Test Paragraphs
Business	500	200 (100+100)	50 (25+25)
Health	500	200 (100+100)	50 (25+25)
Internet	500	200 (100+100)	50 (25+25)
Sports	500	200 (100+100)	50 (25+25)

Let us mention the experimental process for validating empirically the proposed approach to the task of text summarization. We collect the sample paragraphs which are labeled with summary or non-summary in each of the four topics: Business, Sports, Internet, and Health, and encode them into numerical vectors and graphs. For each of 50 examples, the KNN computes its similarities with the 200 training examples, and selects the three similarity training examples as its nearest neighbors. This set of experiments consists of the four independent binary classifications each of in which each paragraph is classified into one of the two labels by the two versions of KNN algorithm. We compute the classification accuracy by dividing the number of correctly classified test examples by the number of test examples, for evaluating the both versions.

In Figure 11, we illustrate the experimental results from deciding whether each paragraph is a summary, or not, using the both versions of KNN algorithm. The y-axis indicates the accuracy which is the rate of the correctly classified examples in the test set. Each group in the x-axis means the domain within which the text summarization which is viewed as a binary classification is performed, independently. In each group, the gray bar and the black bar indicate the accuracies of the traditional version and the proposed version of the KNN algorithm. The most right group in Figure 1 consists of the averages over the accuracies of the left four groups, and the input size which is the dimension of numerical vectors is set to 50.

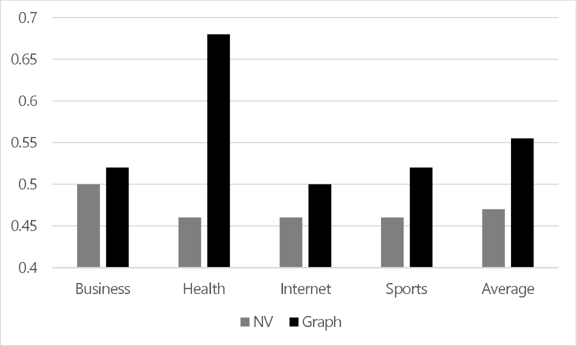

Figure 11. Results from Summarizing Texts in Text Collection: NewsPage.com

Let us make the discussions on the results from doing the text summarization, using the both versions of KNN algorithm, as shown in Figure 11. The accuracy which is the performance measure of this classification task is in the range between 0.46 and 0.67. The proposed version of KNN algorithm works strongly better in the all domains. Furthermore, it shows its strongest results in domain, Health. From this set of experiments, we conclude the proposed version works much better than traditional one, in averaging over the four cases.

B. Opinopsis

This section is concerned with the set of experiments for validating the better performance of the proposed version on the collection, Opinosis. We view the text summarization into a binary classification where each paragraph is classified into summary or non-summary, and collect the paragraphs, labeling manually with one of summary and non-summary from the collection. Each paragraph is exclusively classified into one of the two labels. We fix the input size to 50 and use the accuracy as the evaluation measure. In this section, we observe the performance of the both versions of KNN algorithm, in the three experiments as many as topics.

In Table II, we specify the text collection, Opinosis, which is used in this set of experiments. The collection was used in previous works for evaluating the approaches to text categorization. We extracted the 50 paragraphs in each topic, and they are labeled with 'summary' or 'non-summary', keeping the complete balance over the labels. The 50 paragraphs is partitioned into the 40 as the training set and the 10 as the test set, in each topic. Each text is segmented into paragraphs by the carriage return, and some of them are corrected, in the processing of extracting paragraphs from texts.

Table II
THE NUMBER OF TEXTS AND PARAGRAPHS IN OPINIOPSIS

Category	#Texts	#Training Paragraphs	#Test Paragraphs
Car	23	40 (20+20)	10 (5+5)
Electronic	16	40 (20+20)	10 (5+5)
Hotel	12	40 (20+20)	10 (5+5)

We perform this set of experiments by the process which is described in section IV-A. We collect sample paragraphs which are labeled with 'summary' and 'non-summary' in each of the three domains: 'Car', 'Electronics', and 'Hotel', and we encode them into 50 sized numerical vectors and graphs. For each test example, the both versions of KNN computes its similarities with the 40 training examples and select the three most similar training examples as its nearest neighbors. Each test example is classified into 'keyword' or 'non-keyword' by the two versions of KNN algorithm; we performed the three independent experiments as many as the domains. The classification accuracy is computed by the number of correctly classified test examples by the number of the test examples for evaluating the both versions of KNN algorithm.

In Figure 12, we illustrate the experimental results from the text summarization which is mapped into a classification task, using the both versions of KNN algorithm. Like Figure 11, the y-axis indicates the value of accuracy, and the x-axis indicates the group of two versions by a domain of Opniopsis. In each group, the gray bar and the black bar indicate the results of the traditional version and the proposed version of KNN algorithm. In Figure 12, the most right group indicates the averages of the both version over their results of the left three groups. Therefore, Figure 12 shows the results from classifying paragraphs into one of 'summary', and 'non-summary', by the both versions.

We discuss the results from doing the text summarization which is mapped into a binary classification, using the both versions of KNN algorithm on Opinosis, shown in Figure 12. While the accuracy values of the traditional version stay at 0.5, those of the proposed version range between 0.5 and 0.7. The proposed version works better than the traditional one, in the domain, Hotel. However, it is leaded by the

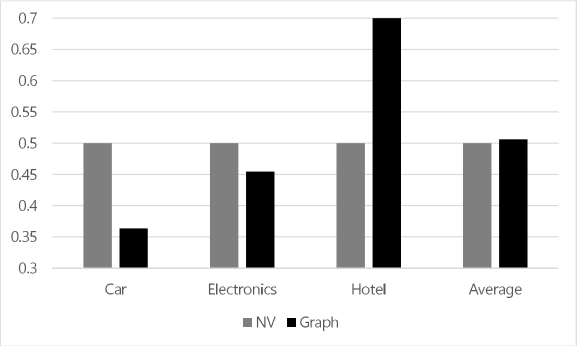

Figure 12. Results from Summarizing Texts in Text Collection: Opiniopsis

traditional one in the others. In spite of that, from this set of experiments, we conclude that the proposed one works competitively with the traditional one in averaging the three cases.

C. 20NewsGroups I: General Version

This section is concerned with one more set of experiments for validating the better performance of the proposed version on text collection, 20NewsGroup I. We gather paragraphs which are labeled with 'summary' or 'non-summary', from each broad category of 20NewsGroups I, by viewing the text summarization into a binary classification. The task of this set of experiments is to classify each paragraph exclusively into one of the two labels in each topic which is called domain. We fix the input size to 50 in encoding the paragraphs and use the accuracy as the evaluation measure. Therefore, in this section, we observe the performances of the both versions in the four different domains.

In Table III, we specify the general version of 20NewsGroups which is used for evaluating the two versions of KNN algorithm. In 20NewsGroup, the hierarchical classification system is defined with the two levels; in the first level, the six categories, alt, comp, rec, sci, talk, misc, and soc, are defined, and among them, the four categories are selected, as shown in Table III. In each category, we extract 250 paragraphs from 4000 or 5000 texts; the first half is labeled with 'summary', and the other half is labeled with 'non-summary'. The 250 paragraphs is partitioned into the 200 ones in the training set and the 50 ones in the test sets, as shown in Table III. In the process of gathering the classified paragraphs, each of them is labeled manually into one of the two categories by scanning individual texts.

Table III
THE NUMBER OF TEXTS AND PARAGRAPHS IN 20NEWSGROUPS I

Category	#Texts	#Training Paragraphs	#Test Paragraphs
Comp	5000	200 (100+100)	50 (25+25)
Rec	4000	200 (100+100)	50 (25+25)
Sci	4000	200 (100+100)	50 (25+25)
Talk	4000	200 (100+100)	50 (25+25)

The experimental process is identical is that in the previous sets of experiments. We collect the paragraphs by labeling manually them with 'summary' or 'non-summary' by scanning individual texts in each of the four domains, comp, rec, sci, and talk, and encode them into numerical vectors and graphs with the input size fixed to 50. For each test example, we compute its similarities with the 200 training examples, and select the three similar ones as its nearest neighbors. The versions of KNN algorithm classify each of the 50 test examples into one of the two categories by voting the labels of its nearest neighbors. Therefore, we perform the four independent set of experiments as many as domains, in each of which the two versions are compared with each other in the binary classification task.

In Figure 13, we illustrate the experimental results from deciding whether each paragraph is a summary, or not, on the broad version of 20NewsGroups. Figure 13 has the identical frame of presenting the results to those of Figure 11 and 12. In each group, the gray bar and the black bar indicates the achievements of the traditional version and the proposed version of KNN algorithm, respectively. In the x-axis, each group indicates the domain within which each paragraph is classified into 'summary', or 'non-summary'. This set of experiments consists of the four binary classifications in each of which it is done so.

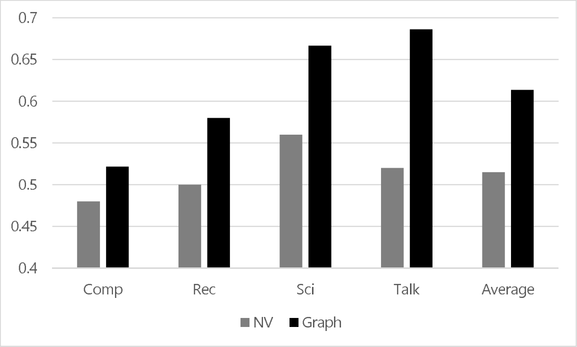

Figure 13. Results from Summarizing Texts in Text Collection: 20NewsGroup I

Let us discuss the results from doing the text summarization using the both versions of KNN algorithm as shown in Figure 13. The accuracies of both versions range between 0.48 and 0.70. The proposed version shows its better performances in three of the four domains. It shows its outstanding difference from the traditional version in the domain, talk. From this set of experiments, the proposed version wins over the traditional one, certainly, in averaging its achievements of the four domains.

D. 20NewsGroups II: Specific Version

This section is concerned with one more set of experiments where the better performance of the proposed version is validated on another version of 20NewsGroups. From each specific topic, separately, we gather the paragraphs which are labeled with 'summary' or 'non-summary'. In this set of experiments, we view the text summarization into a binary classification, and carry out the four binary classifications, independently of each other. We fix the input size of representing paragraphs to 50 and use the accuracy as the evaluation metric. Therefore, in this section, we observe the performances of the both versions of KNN algorithm in the four different domains.

In Table IV, we specify the specific version of 20NewsGroups which is used as the test collection, in this set of experiments. Within the general category, sci, we predefine the four categories: 'electro', 'medicine', 'script', and 'space'. In each topic, we extract 250 paragraphs from approximately 1000 texts and label each of them with 'summary' or 'non-summary', maintaining the complete balance. The set of 250 paragraphs is partitioned into the training set of 200 ones and the test set of 50 ones, as shown in Table IV. We use the accuracy as the metric for evaluating the results from classifying paragraphs.

Table IV
THE NUMBER OF TEXTS AND PARAGRAPHS IN 20NEWSGROUPS II

Category	#Texts	#Training Paragraphs	#Test Paragraphs
Electro	1000	200 (100+100)	50 (25+25)
Medicine	1000	200 (100+100)	50 (25+25)
Script	1000	200 (100+100)	50 (25+25)
Space	1000	200 (100+100)	50 (25+25)

The process of doing this set of experiments is same to that in the previous sets of experiments. We gather sample paragraphs which are labeled with 'summary' or 'non-summary', in each of the four domains: 'electro', 'medicine', 'script', and 'space', and encode them with the fixed input size: 50. We use the two versions of KNN algorithm for their comparisons. Each test paragraph is classified into one of the labels in each domain. We use the accuracy as the evaluation metric.

We present the experimental results from classifying the paragraphs using the both versions of KNN algorithm on the specific version of 20NewsGroups. The frame of illustrating the classification results is identical to the previous ones. In each group, the gray bar and the black bar stand for the achievements of the traditional version and the proposed version, respectively. The y-axis in Figure 14, indicates the classification accuracy which is used as the performance metric. In this set of experiments, we execute the four independent classification tasks which correspond to their own domains, where each paragraph is classified into 'summary' or 'non-summary'.

Let us discuss the results from classifying the paragraphs using the both versions of KNN algorithm on the specific version of 20NewsGroups, as shown in Figure 14. The accuracies as the performance metrics of this classification task which is mapped from the text summarization range between 0.46 and 0.74. The proposed version shows its better results in two of the four domains. It maintain its matching results in the domain, 'space', but is leaded in the domain, 'script'. From this set of experiments, it is concluded that the proposed version have its better performance by averaging over the accuracies of the four domains.

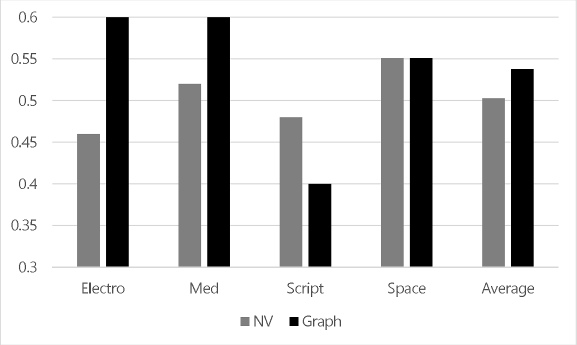

Figure 14. Results from Summarizing Texts in Text Collection: 20NewsGroup II

V. CONCLUSION

Let us discuss the results from summarizing texts using the two versions of KNN algorithm. In these sets of experiments, we compare the two versions with each other in the classification tasks which is mapped from the text summarizations. The proposed version shows its better results in all of the four collections. The classification accuracies of the traditional version range between 0.46 and 0.55, while those of the proposed version range between 0.37 and 0.70. From the four sets of experiments, we conclude that the proposed version improves the text summarization performance, as the contribution of this research.

Let us mention the remaining tasks for doing the further research. We apply and validate the proposed research in summarizing technical documents in specific domains such as medicine or engineering rather than news articles in various domains. We define and characterize more advanced operations mathematically on graphs which represent texts. We modify more advanced machine learning algorithms into their graph based version, using the more sophisticated operations. We implement the text summarization system as a system module or an independent software by adopting the proposed approach.

REFERENCES

[1] N.F. Noy and C. D. Hafner, "State of the Art in Ontology Design", AI Magazine, Vol 18, No 3, 1997.

[2] T. Jo, "NeuroTextCategorizer: A New Model of Neural Network for Text Categorization", 280-285, The Proceedings of ICONIP 2000.

[3] T. Jo and N. Japkowicz, "Text Clustering using NTSO", pp558-563, The Proceedings of IJCNN, 2005.

[4] T. Jo, “The Implementation of Dynamic Document Organization using Text Categorization and Text Clustering” PhD Dissertation of University of Ottawa, 2006.

[5] Y. Zheng, X. Cheng, R. Huang, and Y. Man, “A comparative study on text clustering methods”, 644-651, Advanced Data Mining and Applications, 2006.

[6] T. Jo, M. Lee, and T. M Gatton, “Modifying a Kernel based Learning in Text Categorization using an Inverted Index based Operation”, 387-391, The Proceedings of International Conference on Information and Knowledge Engineering, 2007.

[7] T. Jo and M. Lee, “Kernel based Learning Suitable for Text Categorization”, 289-294, The Proceedings of 5th IEEE International Conference on Software Engineering Research, Management and Applications, 2007.

[8] T. Jo, “Single Pass Algorithm for Text Clustering by Encoding Documents into Tables”, 1749-1757, Journal of Korea Multimedia Society, Vol 11, No 12, 2008.

[9] T. Jo, “Modified Version of SVM for Text Categorization”, 52-60, International Journal of Fuzzy Logic and Intelligent Systems, Vol 8, No1, 2008.

[10] T. Jo, “Neural Text Categorizer for Exclusive Text Categorization”, 77-86, Journal of Information Processing Systems, Vol 4, No 2, 2008.

[11] T. Jo and D. Cho, “Index Based Approach for Text Categorization”, 127-132, International Journal of Mathematics and Computers in Simulation, Vol 2, No 1, 2008.

[12] T. Jo, “NTSO (Neural Text Self Organizer): A New Neural Network for Text Clustering”, 31-43, Journal of Network Technology, Vol 1, No 1, 2010.

[13] T. Jo, “NTC (Neural Text Categorizer): Neural Network for Text Categorization”, 83-96, International Journal of Information Studies, Vol 2, No 2, 2010.

[14] D. Allemang and J. Hendler, “Semantic Web for the Working Ontologies”, Mrgan Kaufmann, 2011.

[15] K. Abainia, S. Ouamour, and H. Sayoud. “Neural Text Categorizer for topic identification of noisy Arabic Texts”, 1-8, Proceedings of 12th IEEE Conference on Computer Systems and Applications, 2015.

[16] T. Jo, “Normalized Table Matching Algorithm as Approach to Text Categorization”, 839-849, Soft Computing, Vol 19, No 4, 2015.

[17] T. Jo, “Graph based KNN for Content based Word Classification”, 24-29, The Proceedings of 12th International Conference on Multimedia Information Technology and Applications, 2016.

[18] L. Vega and A. Mendez-Vazquez, “Dynamic Neural Networks for Text Classification”, 6-11, The Proceedings of International Conference on Computational Intelligence and Applications, 2016.

[19] L.R. Flaih “Web page Classification by Using PCA and Neural Network”, 242-256. Cihan University-Erbil Scientific Journal, Vol 1, No 1, 2017.

[20] T. Jo, “K Nearest Neighbor specialized for Word Categorization in Current Affairs by Graph based Version”, 64-65, The Proceedings of 1st International Conference on Advanced Engineering and ICT-Convergence, 2018.

[21] T. Jo, “Comparing Graph based K Nearest Neighbor with Traditional Version in Word Categorization in NewsPage.com, 12-18, International Journal of Advanced Social Sciences, Vol 1, No 1, 2018.

[22] T. Jo, “Graph based KNN for Text Categorization”, 260-264, The Proceedings of IEEE 18th International Conference on Advanced Communication Technology, 2018.

[23] T. Jo, “Graph based KNN for Text Summarization”, 438-442, The Proceedings of IEEE 18th International Conference on Advanced Communication Technology, 2018.

[24] T. Jo, “Clustering Texts using Feature Similarity based AHC Algorithm”, 5993-6003, Journal of Intelligent and Fuzzy Systems, Vol 35, 2018.

[25] T. Jo, “Semantic Word Categorization using Feature Similarity based K Nearest Neighbor”, 67-78, Journal of Multimedia Information Systems, 2018.

[26] T. Jo, “Improving K Nearest Neighbor into String Vector Version for Text Categorization”, 1091-1097, ICACT Transaction on Communication Technology, Vol 7, No 1, 2018.

[27] T. Jo, “Automatic Text Summarization using String Vector based K Nearest Neighbor”, 6005-6016, Journal of Intelligent and Fuzzy Systems, Vol 35, 2018.

[28] T. Jo, “Modification of K Nearest Neighbor into String Vector based Version for Classifying Words in Current Affairs”, 72-75, The Proceedings of International Conference on Information and Knowledge Engineering, 2018.

[29] T. Jo, “Improving K Nearest Neighbor into String Vector Version for Text Categorization”, 1091-1097, ICACT Transaction on Communication Technology, Vol 7, No 1, 2018.

[30] T. Jo, “Automatic Text Summarization using String Vector based K Nearest Neighbor”, 6005-6016, Journal of Intelligent and Fuzzy Systems, Vol 35, 2018.

[31] T. Jo, “String Vector based AHC Algorithm for Word Clustering from News Articles”, 83-86, The Proceedings of International Conference on Information and Knowledge Engineering, 2018.

[32] T. Jo, “Graph based Version of K Nearest Neighbor for classifying News Articles”, 4-7, The Proceedings of International Conference on Green and Human Information Technology Part I, 2019.

[33] T. Jo, “Validation of Graph based K Nearest Neighbor for Summarizing News Articles”, 5-8, The Proceedings of International Conference on Green and Human Information Technology Part II, 2019.

[34] T. Jo, “Text Classification using Feature Similarity based K Nearest Neighbor”, 13-21, AS Medical Science, Vol 3, No 4, 2019.

[35] T. Jo, “Introduction of String Vectors to AHC Algorithm for Clustering News Articles”, 150-153, The Proceedings of 21st International Conference on Artificial Intelligence, 2019.

[36] T. Jo, “Graph Similarity Metric for Modifying K Nearest Neighbor for Classifying Texts”, unpublished, 2020.

[37] T. Jo, “Semantic String Operation for Specializing AHC Algorithm for Text Clustering”, 10472-019-09687-x, Annals of Mathematics and Artificial Intelligence, 2020.

[38] T. Jo, “Semantic String Operation for Specializing AHC Algorithm for Text Clustering”, 10472-019-09687-x, Annals of Mathematics and Artificial Intelligence, 2020.