Topic based Segmentation using K Nearest Neighbor modified by Graph Similarity Metric

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 segmentation. The graph is more graphical for representing a word and the text segmentation is able to be viewed into a binary classification where each adjacent paragraph pair is classified into boundary or continuance. In the proposed system, a list of adjacent paragraph pairs is generated by sliding a text with the two sized window, each pair is classified by the proposed KNN version, and the boundary is put between the pairs which are classified into boundary. The proposed KNN version is empirically validated as the better approach in deciding whether each pair should be separated from each other or not in news articles and opinions. In this article, an adjacent paragraph pair is encoded into a weighted and undirected graph and it is represented into a list of edges.**

I. Introduction

Text segmentation refers to the process of making the mark in the point where one topic is transitioned into another topic. Each text is partitioned into paragraphs and pairs of adjacent paragraphs are encoded into structured forms. We prepare the sample paragraph pairs which are labeled with boundary or non-boundary and construct the classification capacity by learning the sample pairs. From novice texts, we generate pairs of adjacent paragraphs and put boundary into position corresponding to pairs which are classified with the boundary. The assumption underlying in this research is that the text segmentation is viewed as a binary classification, and a supervised learning algorithm is applied as the approach to the task.

Let us mention some points which motivate 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 to the text mining tasks[4]. The graphs which are called ontology or word net became the popular representations of knowledge and information[14][1]. Accordingly, many algorithms were developed and improved for manipulating graphs[]Allemang and Hendler 2011Noy and Hafner 1997. Therefore, by the motivations, we modify the KNN (K Nearest Neighbors) into its graph based version, and apply it to the text segmentation as an instance of text mining.

Let us mention what we propose in this research as some agenda. We encode pairs of adjacency paragraphs into graphs each of which have vertices indicating words and edges indicating their semantic relations. We define the similarity measure between graphs which consists their difference vertices and edges as that between two paragraph pairs. We modify the KNN into its graph based version, using the similarity measure, and apply it to the text segmentation where each paragraph pair is classified into where we put delimiter, or not. Note that the graphs which represent texts belong to the class, 'undirected weighted graphs', and are represented into the adjacency matrices in the implementation level.

Let us mention some benefits which are expected from this research. We expect the better text segmentation performance in using the proposed KNN version, by avoiding the problems from encoding texts into numerical vectors. We expect the more transparency where texts contents are more easily visible only by their representations in the proposed KNN version, since the graphs are more graphical versions than numerical vectors. We expect the more compactness from graphs which represent texts for processing them more efficiently than from numerical vectors. Therefore, the goal of this research is to implement the text segmentation system which has 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-C, we survey previous works, respectively, on the string vector based machine learning algorithms and neural networks. Therefore, in this section, we provide the history about this research, by surveying the relevant previous works.

A. Applications to Related Tasks

This section is concerned with the text segmentation and its related tasks where the modernized KNN algorithm is applied. We mention the text categorization as the base task, and the text summarization and the text segmentation are derived from it. We present the previous cases of applying the modernized KNN algorithm for the text segmentation which is covered in this research. We mention the cases of doing it for the text summarization as a related task, as well as the text segmentation. This section is intended to survey the previous cases of applying the modernized KNN algorithm for the text segmentation and its related tasks.

Let us explore the previous cases of applying the modernized version of the KNN algorithm for the text categorization as the base task for the text segmentation. In 2018, Jo initiated modifying the KNN algorithm into the graph based version as an approach to the text categorization [21]. In 2019, he started to observe its better performance than the traditional version in the text categorization [31]. In 2020, he complete validating the better performance of the graph based version through the three sets of experiments [35]. In the above literatures, we presented the previous cases of applying the graph based version of the KNN algorithm for the text categorization.

Let us survey the cases of applying the modernized KNN algorithm for the text summarization which is derived from the text categorization. It was initially asserted as an approach to the text summarization, by Jo in 2017 [19]. Its better performance than the traditional version was initially discovered in the text summarization by Jo in 2018 [22]. Validating empirically its better performance was finalized, but it is not published, yet [36]. The text summarization is interpreted into a binary classification where each paragraph is classified into summary or non-summary, in the above literatures.

Let us review the previous works where the graph based KNN algorithm was applied for segmenting a text based on its contents. It was initially asserted that the KNN algorithm is modified into the graph based version as the approach to the text segmentation, in 2017 [20]. The graph based KNN algorithm is compared with the traditional version, and its better performance is observed in segmenting texts in a small text collection, in 2019 [32]. This research is aimed to finalize the empirical validation of the better performance through the real experiments. In the above literatures, the text segmentation is interpreted into a binary classification of an adjacent paragraph pair into boundary or continuance.

Let us survey the previous cases of applying the graph based version to the three related tasks. The text segmentation which is covered in this research is the process of partitioning a text into subtexts, based on the paragraph topics; a boundary is set between paragraphs with their topic transition. The modified version of the KNN algorithm which is adopted in this research as the approach to the text segmentation, processes graphs directly. In the above literatures, the proposed KNN version was applied to the text categorization and the text summarization as well as the text segmentation. The research on the graph based KNN version for the text segmentation has progressed, and we need to complete the empirical validations of its more desirability in the task.

B. Encoding Schemes

This section is concerned with the previous works on the schemes of encoding texts into structured data. In this research, we propose that texts should be encoded into graph, for modifying the KNN algorithm as the approach to the text segmentation. In this section, we will survey the cases of encoding texts into numerical vectors, tables, and string vectors. The KNN algorithm and the AHC algorithm are modified into the versions which process such kinds of structured data, directly. This section is intended to survey the previous works on the encoding schemes which are relevant to this research.

Let us explore the previous cases of encoding texts into numerical vectors using the modernized similarity metric. In 2018, texts were encoded into numerical vectors in using the AHC algorithm as the approach to the text clustering [23]. In 2018, words were encoded into numerical vectors, in using the KNN algorithm as the approach to the word categorization [24]. In 2019, texts encoded so in using the KNN algorithm as the approach to the text categorization [33]. The similarity metric which is used in the AHC algorithm and the KNN algorithm is modernized by considering the feature similarities and the feature value similarities in the above literatures.

Let us survey the previous works where texts are encoded into tables for modifying the classification algorithm and the clustering algorithm. In 2008, texts were initially encoded into tables in categorizing texts by Jo and Cho [11]. The online linear clustering algorithm was modified into the table based version as the approach to the text clustering in 2008 [8]. In 2015, the table based matching algorithm with its better performance and its more stability to different domains was proposed as the approach to text categorization [16]. In the above literatures, we present the previous cases of the classification algorithm and the clustering algorithm where texts are encoded into tables.

Let us mention the previous works where texts are encoded into string vectors, instead of numerical vectors. In 2018, texts are encoded into string vectors for modifying the KNN algorithm as the approach to the text categorization [25]. In 2018, the string vector based version of the KNN algorithm was applied to the text summarization which was mapped into the binary classification of texts [26]. In 2020, texts were encoded so for modifying the AHC algorithm as the approach to the text clustering [37]. In the above literatures, we presented the previous cases of encoding texts into string vectors as the structured forms.

We surveyed above the previous works on the structured data into which texts or words are encoded. They are encoded into numerical vectors and the similarity metric which is tolerant to the poor discriminations among sparse vectors is defined. Texts are encoded into tables, and the similarity metric between tables is used for modifying the KNN algorithm and the AHC algorithm into the table based version. Texts are encoded into string vectors, and the similarity metric between them is defined as a semantic operation. In this research, paragraph pairs which are given as texts are encoded into graphs, and the similarity metric between graphs is defined and described in detail in Section 3.2.

C. String Vector based Machine Learning Algorithms

This section is concerned with the previous works on the string vector based machine learning algorithms. A string vector is defined as an ordered set of strings and the machine learning algorithms in the works which are surveyed in this section process string vector, directly. In this section, we mention the three string vector based machine learning algorithms: the string vector based neural networks, the string vector based AHC algorithm, and the string vector based SVM (Support Vector Machine). The significance of the previous works is to try to solve the problems in encoding texts into numerical vectors, such as huge dimensionality and sparse distribution. This section is intended to explore the previous works on the string vector based machine learning algorithms, as non-numerical vector based ones.

Let us survey the previous works on the version of the KNN algorithm which processes string vectors directly, instead of numerical vectors. It was initially proposed as the approach to the word categorization, in 2018 [27]. It was applied to the text categorization, in 2018 [28]. The text summarization is mapped into a binary classification of each paragraph into summary or non-summary, and the version of the KNN algorithm was applied to it, in 2018 [29]. In the above literatures, we present that the string vector based KNN algorithm was proposed and applied to the text mining tasks.

Let us mention the previous works on the string vector based AHC algorithm as a non-numerical vector based clustering algorithm. In 2018, the string vector based AHC algorithm was proposed as the approach to the word clustering [30]. In 2019, it was applied to the text clustering by encoding texts into string vectors [34]. Proceeding the research on the string vector based AHC algorithm was finalized by completing the empirical validation of its better performance than the traditional version, in 2020 [38]. In the above literatures, we present the string vector based AHC algorithm which processes string vectors as the alternatives to numerical vectors.

Let us mention the previous works on the SVM (Support Vector Machine) whose kernel function is the string vector based kernel function. In 2007, the string vector kernel function which is installed in the SVM was initially defined by building the inverted index of strings [6]. The similarity matrix was constructed as the basis for computing the output value of the string kernel function, in 2007 [7]. Its better classification performance than the traditional SVM, the Naive Bayes, and the KNN algorithm which are used as the main approaches to the text categorization, was validated empirically, in 2008 [9]. In the above literatures, we mention the definition of the string kernel function and the similarity metric and the modification of the SVM by the string kernel function.

We surveyed the previous works on the string vector based machine learning algorithms as non-numerical vector based ones, but we propose the graph based version which processes graphs directly, in this research. Raw data or texts are encoded into graphs for using the proposed machine. The KNN algorithm is modified into the version which classifies graphs, using the similarity metric which is described in Section III-B. The modified version of the KNN algorithm is applied for implementing the text segmentation system. The significance of this research is to propose another kind of non-numerical vector based machine learning algorithm and to apply it to the text segmentation which is mapped into the classification task.

D. String Vector based Neural Networks

This section is concerned with the previous works on the neural networks which process string vectors directly. In this research, texts are encoded into graphs as the alternative representation to the string vector and the numerical vector. We mention the NTC (Neural Text Categorizer) which is an approach to the text categorization, and the NTSO (Neural Text Self Organizer) which is an approach to the text clustering. The two neural networks were created as the previous trials of solving the problems in encoding texts into numerical vectors. This section is intended to survey the previous works which cover or cite either of the two neural networks.

Let us survey the previous works on the NTSO which is an approach to the text clustering. The NTSO was initially proposed as the approach to the text clustering by Jo and Japkowicz, in 2005 [3]. The NTSO was mentioned as an innovative neural networks in 2006 [5]. The progress of the research on the NTSO was finalized by validating empirically its better clustering performance, in 2010 [12]. In the above literatures, we present the proposal, the citation, and the validation of the NTSO.

Let us explore the previous works on the NTC (Neural Text Categorizer) as another 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 the weight updating process, in 2008 [10]. Its better classification performance than the KNN, the Na?ve Bayes, and the SVM, was 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 as the approach to the text classification.

Let us review the previous works which use or cite the NTC. It was proposed by Jo and used for classifying Arabian texts by Abainia et al. in 2015 [15]. It was mentioned as an innovative approach to the text clustering by Vega and Mendez-Vazquez, in 2016 [17]. It was mentioned in applying the neural networks to the web page classification with the PCA (Principal Component Analysis) by Flaih, in 2017 [18]. In the above literatures, we present the citation and the application of the NTC.

We explored the previous works on the two string vector based neural networks: NTC and NTSO. The former is used for classifying texts as a supervised neural networks, and the latter is used for clustering texts as an unsupervised neural networks. It is possible to apply the NTC and the NTSO, respectively for classifying words and clustering them, as well as texts. It is also possible to transition the NTC and the NTSO between the supervised learning and the unsupervised learning. In next research, we consider applying the NTC and the NTSO for text segmentation which is covered in this research.

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 segmentation, 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 segmentation. 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 transforming a text into a graph. A graph is defined into two sets, the vertex set and the edge set, in the context of the data structures. A vertex is given as a word, and an edge is given as a semantic similarity between words, in the graph which represents a text. A graph is assumed to an edge set for computing a similarity between graphs. This section is intended to describe each step of encoding a text into a graph.

Figure 1 illustrates that the k words are given as the vertices from a single text. The words in it are given as vertices in representing a text into a graph. From the single text which is given as in the left side of Figure 1, k words which are given in the right side are generated. The steps of indexing a text into a list of words are the tokenization, the stemming, and the stopword removal. A vertex set is constructed in this step, by indexing the text with the three steps.

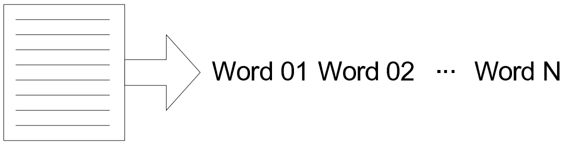

Figure 1. Text Indexing

The construction of the similarity matrix with the vertices for defining edges in the graph is illustrated in Figure 2. The N words are gathered as the vertices by the previous process which is shown in Figure 1. The similarity matrix where its rows and columns correspond to the N words, and each element is given as a similarity between words is constructed for defining edges by computing a similarity for each pair. In the similarity matrix which is presented in Figure 2, its off-diagonal elements are given as normalized values between zero and one, and its diagonal ones are given as ones. The threshold between zero and one is given as an external parameter for selecting some edges.

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

Figure 2. Similarity Matrix

A simple example of the graph which represents a text is illustrated in Figure 3. The four words, information, computer, business, and system, are set as the vertices. The similarities among them are computed based on their collocations in the corpus, and they are defined as edges. The four words in Figure 3 are linked completely; edges are all possible pairs of the four words. Only some need to be selected among them for more efficient processing.

Let us make some remarks on the process of mapping a text into a graph. In the context of the data structure and the graph theory, the graph is defined as the vertex set and the edge set, formally. Words in the text are given as the vertices, and the semantic similarities among them are given as the edges, in representing a text into a graph. A graph is

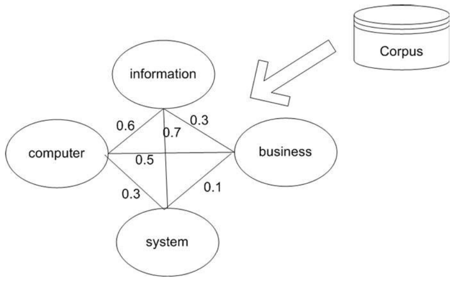

Figure 3. Graph representing a Text

viewed as an edge set in implementing the proposed KNN algorithm. We need to take only some edges, rather than the complete links for process graphs more efficiently.

B. Similarity between Two Graphs

This section is concerned with the similarity metric between two graphs which represent texts. In the previous section, we explained the process of encoding texts into graphs. A graph is viewed as an edge set, the similarity metric between edges is defined as the base, and it is expanded into one between graphs. The similarity between graphs is always given as a normalized value, and the similarity metric is used for modifying the KNN algorithm into the version which processes graphs, directly. This section is intended to describe the similarity metric 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.

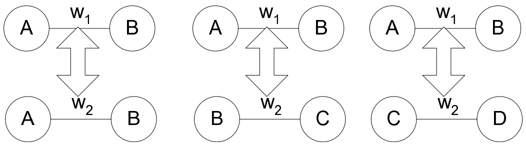

Figure 4. Three Cases in computing Edge Similarity

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 metric between graphs which is described in this section. A graph is represented into a set of edges, and the computation starts with the similarity between two edges. It is expanded into one between two graphs by means of one between an edge and a graph. The similarity metric is utilized for modifying the KNN algorithm into the graph based version as the approach to the text segmentation. In future, we need to define more operations on graphs for modifying other machine learning algorithms.

C. Proposed Version of KNN

This section is concerned with the graph based KNN algorithm, as the approach to the text segmentation. In this previous section, we described the similarity metric between two graphs, under the assumption of each graph which is viewed as an edge set. The similarity metric is used for computing the similarities of a novice item which is represented into a graph with the training graphs, in the proposed KNN algorithm. We will adopt the proposed version for implementing the text segmentation system. This section is intended to describe the proposed version of the KNN algorithm which classifies graphs, 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.

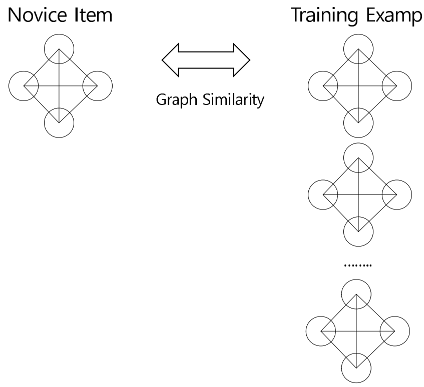

Figure 5. Similarities of a Novice Graph with Sample Ones

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),

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

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}.

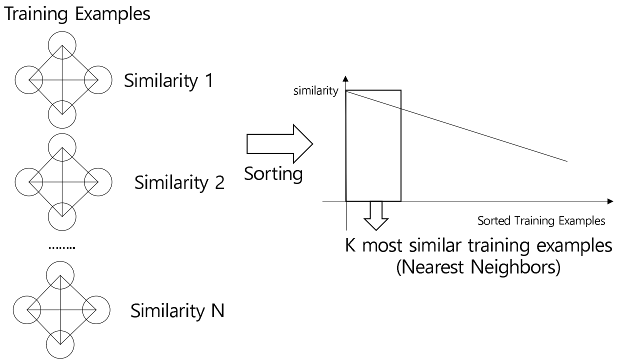

Figure 6. Selection of Nearest Neighbors from Training Examples

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),

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

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

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

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),

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

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.

Let us make some remarks on the proposed version of the KNN algorithm which classifies graphs directly as

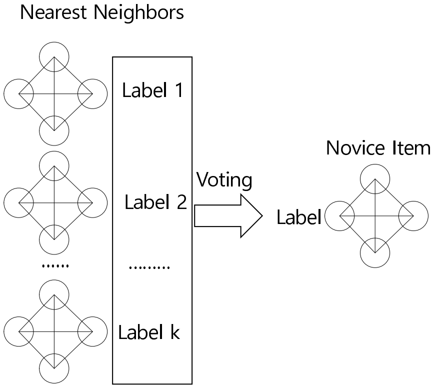

Figure 7. Voting Labels of Training Examples for deciding One of Novice Example

the approach to the text segmentation. Texts are encoded into graphs, instead of numerical vectors, for using the proposed version of the KNN algorithm. The similarity metric between graphs is defined and used for computing the similarities of a novice item with the sample ones. The sample items are ranked by their similarities with the novice item, and the K sample items are selected with their highest similarities as the nearest neighbors. The labels of the nearest neighbors are voted for deciding one of the novice item as the classification process.

D. Text Segmentation System

This section is concerned with the system architecture and the execution process of the text segmentation system. The text segmentation is viewed into the binary classification of each paragraph pair, and the KNN algorithm which is described in Section III-C is applied. Adjacent paragraph pairs are generated by sliding a text with the two sized window, and classified into continuance or boundary. A boundary between subtexts is put between each adjacent paragraph pair which is classified into boundary, and more than one subtext is generated as output of the system. This section is intended to describe the system architecture and the execution process which are needed for designing the text segmentation system.

Sampling the paragraph pairs which are labeled manually with boundary and continuance and classifying a novice one into one of the two categories is illustrated in Figure 8. Because a same paragraph pair is classified differently depending on its domain, the task is called domain dependent task. The sample paragraph pairs which are labeled with boundary or continuance are collected in each domain. Adjacent paragraph pairs are taken from the input text, and each of them is classified into one of the two categories. The task should be distinguished from the text classification which is a domain independent classification, where a same item is always classified identically.

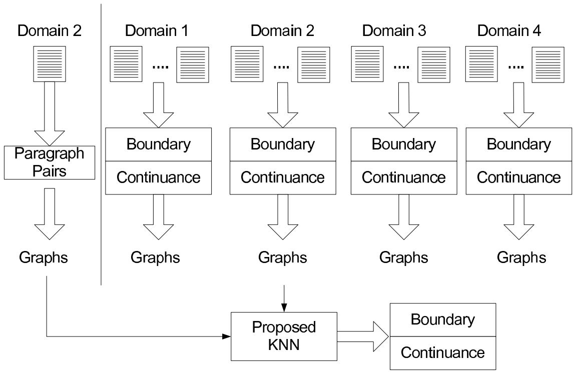

Figure 8. Collecting Sample Paragraph Pairs

The system architecture of the text segmentation system is illustrated in Figure 9. The text partition & sliding module partitions an input text into paragraphs, and takes adjacent partition pairs by sliding the two sized window on them, and the encoder module encodes them into graphs. The similarity computation module computes the similarities of a novice graph with the sample graphs, and takes the nearest neighbors, depending on their similarities. The voting module votes the labels of the nearest neighbors, in order to decide label of the novice one. The adjacent paragraph pairs which are taken from the input text are classified into boundary or continuance, and the boundary is marked on the point between paragraphs in each pair which is classified into boundary in the system.

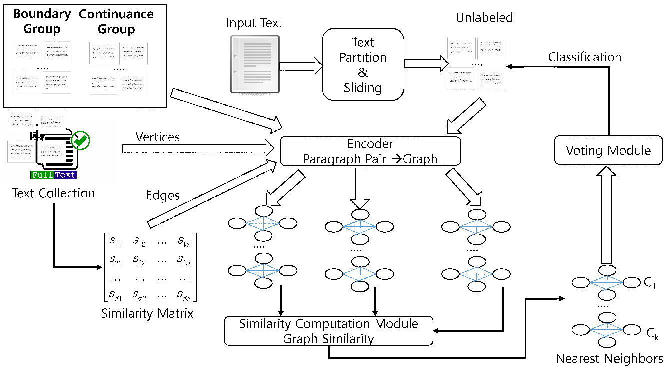

Figure 9. System Architecture

The execution flow of the text segmentation system is illustrated in Figure 10. Texts which belong to an identical domain are collected and the paragraph pairs which are manually labeled with boundary and continuance are gathered as the samples. Adjacent paragraph pairs are generated from the input text, and they are encoded into graphs, together with the samples. They are classified into boundary or continuance by the KNN algorithm which was described in Section III-C. A boundary is marked between paragraph in each pair which is classified into boundary, and the input text is partitioned into subtexts which are extracted as the final output of this system.

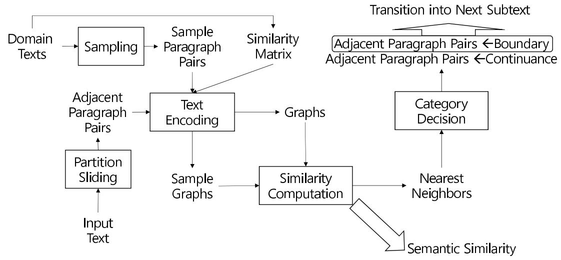

Figure 10. Execution Process

Let us make some remarks on the system architecture and the execution flow of the text segmentation system which is presented in Figure 9 and 10. The text segmentation is mapped into a binary classification of paragraph pairs, and encoding texts into graphs and the similarity between them are proposed in this research. The KNN algorithm is modified into the graph based version, using the similarity between graphs as the approach to the text segmentation. The system architecture and the execution flow are provided in this research and needed for doing the only general design. In the next research, we will provide the detail design and the implementation of the system.

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 segmentation on the collection, NewsPage.com. In Section IV-B, we show the results from applying it for classifying paragraph pairs into boundary or continuance, 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 segmentation 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 segmentation into the binary classification where each adjacent paragraph pair is classified into boundary and continuance, and, by sliding window on paragraphs of each text, gather the paragraph pairs which are labeled with one of the two categories, from the collection, topic by topic. Each paragraph pair is classified exclusively into one of the two labels. We fix the input size as 50 in encoding paragraph pairs into numerical vectors and string 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 adjacent paragraph pairs and label them with boundary or continuance, keeping the complete balance over the two labels. In each category, the set of 250 paragraph pairs is partitioned into the training set of 200 ones and the test set of 50 ones. Each text is segmented into paragraphs by a carriage return, and adjacent paragraph pairs are generated by sliding two sized window on the list of paragraphs.

Table I
The Number of Texts and Paragraph Pairs in NewsPage.com

Category	#Texts	#Training Pairs	#Test Pairs
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 segmentation. We collect the sample paragraphs which are labeled with boundary or continuance 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 classifying each adjacent paragraph pair into boundary or continuance, 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 11 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.

Let us make the discussions on the results from doing

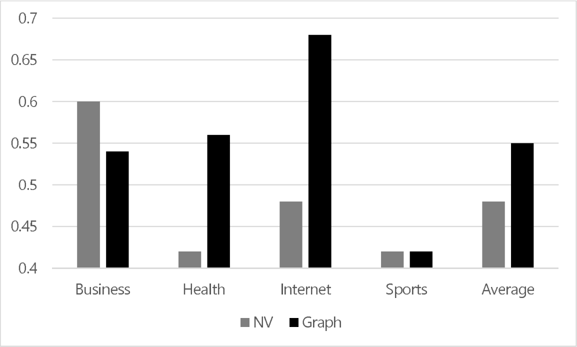

Figure 11. Results from Segmenting Texts in Text Collection: NewsPage.com

the text segmentation, 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 0.4 and 0.67. The proposed version of KNN algorithm works strongly better in the two domains, Health and Internet. It matches in domain, Sports, but loses in the domain, Business. In spite of that, from this set of experiments, we conclude the proposed version works 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 segmentation into a binary classification where each adjacent paragraph pair is classified into boundary or continuance, and collect the paragraphs pairs, sliding paragraphs in each text by two sized window and labeling manually with one of boundary and continuance from the collection. Each paragraph pair 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 test collection is used in previous works for evaluating approaches to text categorization. We extract the 50 adjacent paragraph pairs in each topic, and label them with 'boundary' or 'continuance', keeping the complete balance. The set of 50 paragraph pairs is portioned into the 40 as the training set and the 10 as the test set, in each topic. In the process of generating the paragraph pairs, each text is segmented into paragraphs by the carriage return, the adjacent paragraph pairs are generated by sliding the paragraphs.

We perform this set of experiments by the process which is described in section IV-A. We collect sample adjacent paragraph pairs which are labeled with 'boundary' and 'continuance' 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 'boundary' or 'continuance' 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.

Table II
THE NUMBER OF TEXTS AND PARAGRAPH PAIRS IN OPINIOPSIS

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

In Figure 12, we illustrate the experimental results from the text segmentation 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 adjacent paragraph pairs into one of 'boundary', and 'continuance', by the both versions.

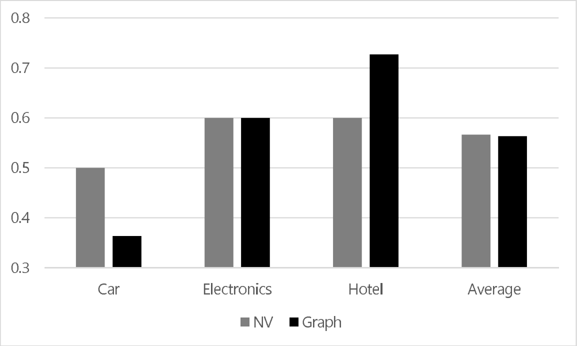

Figure 12. Results from Segmenting Texts in Text Collection: Opiniopsis

We discuss the results from doing the text segmentation which is mapped into a binary classification, using the both versions of KNN algorithm, shown in Figure 12. The accuracy values of the both versions range between 0.35 and 0.75. The proposed version works better than the traditional one in the domain, Hotel. It is comparable with the traditional version in the domain, Electronics and leaded in Car. 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 adjacent paragraph pairs which are labeled with 'boundary' or 'continuance', from each broad category of 20NewsGroups I, by viewing the text segmentation into a binary classification. The task of this set of experiments is to classify each paragraph pair exclusively into one of the two labels in each topic which is called domain. We fix the input size to 50 in encoding paragraph pairs 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 adjacent paragraph pairs from 4000 or 5000 texts; the first half is labeled with 'boundary', and the other half is labeled with 'continuance'. The 250 paragraphs pairs 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 paragraph pairs, each of them is labeled manually into one of the two categories by scanning individual texts.

Table III
THE NUMBER OF TEXTS AND PARAGRAPH PAIRS IN 20NEWSGROUPS I

Category	#Texts	#Training Pairs	#Test Pairs
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 adjacent paragraph pairs by labeling manually them with 'boundary' or 'continuance' 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 we put a boundary, or not, between two adjacent paragraphs, 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 pair is classified into 'boundary', or 'continuance'. This set of experiments consists of the four binary classifications in each of which it is done so.

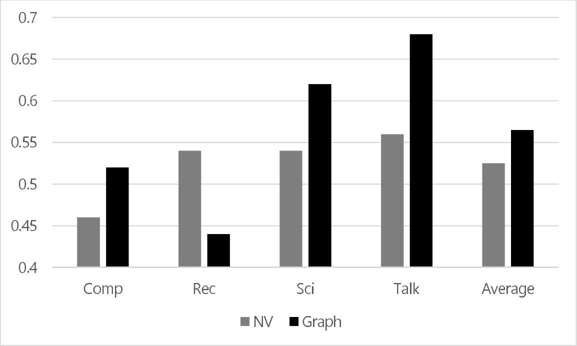

Figure 13. Results from Segmenting Texts in Text Collection: 20NewsGroup I

Let us discuss the results from doing the text segmentation using the both versions of KNN algorithm as shown in Figure 13. The accuracies of both versions range between 0.45 and 0.7. 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. However, its performance is leaded in the domain, rec. From this set of experiments, the proposed version wins over the traditional one, 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 adjacent paragraph pairs which are labeled with 'continuance' or 'boundary'. In this set of experiments, we view the text segmentation into a binary classification, and carry out the four binary classifications, independently of each other. We fix the input size of representing the paragraph pairs 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 adjacent paragraph pairs from approximately 1000 texts and label each of them with 'boundary' or 'continuance', maintaining the complete balance. The set of 250 paragraph pairs 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 them.

Table IV
THE NUMBER OF TEXTS AND PARAGRAPH PAIRS IN 20NEWSGROUPS II

Category	#Texts	#Training Pairs	#Test Pairs
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 paragraph pairs which are labeled with 'boundary' or 'continuance', 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 pair 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 paragraph pairs 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 pair is classified into 'boundary' or 'continuance'.

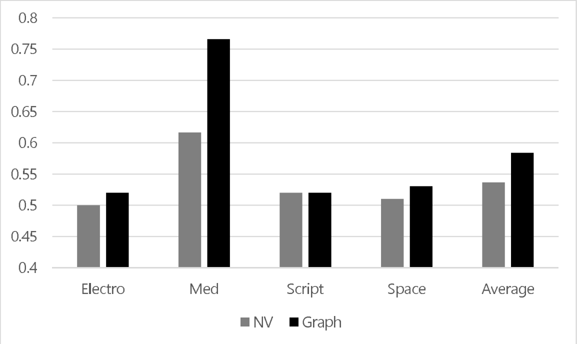

Figure 14. Results from Segmenting Texts in Text Collection: 20NewsGroup II

Let us discuss the results from classifying the adjacent paragraph pairs 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 segmentation range between 0.45 and 0.76. The proposed version shows its better results in three of the four domains. It keeps its matching results 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.

V. CONCLUSION

Let us discuss the results from segmenting a text 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 segmentations. The proposed version shows its better results in all of the four collections. The classification accuracies of the traditional version range between 0.41 and 0.62, while those of the proposed version range between 0.44 and 0.76. From the four sets of experiments, we conclude that the proposed version improves the text segmentation 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 segmenting each technical document based on its contents 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 segmentation system as a system module or an independent program 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", 558-563, 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", pp644-651, 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”, pp839-849, Soft Computing, Vol 19, No 4, 2015.

[17] 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.

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

[19] T. Jo, “K Nearest Neighbor for Text Summarization using Feature Similarity”, DOI: 10.1109/ICCCCEE.2017.7866705, Proceedings of International Conference on Communication, Control, Computing and Electronics Engineering, 2017.

[20] T. Jo, “Graph based KNN for Text Segmentation”, 322-327, The Proceedings of Computer Science and Computational Intelligence, 2017.

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

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

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

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

[25] 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.

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

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

[32] T. Jo, “Specializing K Nearest Neighbor for Content based Segmentation of News Article by Graph Similarity Metric”, 9-12, The Proceedings of International Conference on Green and Human Information Technology Part II, 2019.

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

[34] 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.

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

[36] T. Jo, “Summarizing Texts Automatically by Graph based Version of K Nearest Neighbor”, unpublished, 2018.

[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.