Graph Similarity Metric for Modifying K Nearest Neighbor for Classifying Texts

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 categorization. The graph is more graphical for representing a word and the synergy effect between the text categorization and the word categorization is expected by combining them with each other. In this research, we propose the similarity metric between two graphs representing words, modify the KNN algorithm by replacing the exiting similarity metric by the proposed one, and apply it to the text categorization. The proposed KNN is empirically validated as the better approach in categorizing texts in news articles and opinions. In this article, a word is encoded into a weighted and undirected graph and it is represented into a list of edges.**

I. Introduction

Text categorization refers to the process of classifying each text into its relevant topics or categories among the predefined ones. As its preliminary tasks, a finite number of categories are predefined and sample texts which are labeled with one or some of the predefined are prepared. As the learning process, using the sample labeled texts, the classification capacity is constructed. Subsequent texts which are given as ones separated from the sample labeled texts are classified as the generalization process. Even if other kinds of approaches such as manual rule based schemes and other heuristic ones are available, in this research, we assume that the supervised learning algorithms are used as the approach.

Let us mention some points which provide the motivations for doing this research. Encoding texts into numerical vectors causes problems such as huge dimensionality and sparse distribution [3]. The graphs became the popular representations of knowledge or information which are called ontologies or word nets [1][17]. Because the ontologies are used for representing knowledge as graphs, in previous works, many algorithms for manipulating graphs. Therefore, by these motivations, we encode texts into graphs, and modify the machine learning algorithms into versions which receive graphs as input data.

Let us mention what we propose in this research as its ideas. Instead of a numerical vector, each text is encoded into the graph where its vertices are words and its edges are the semantic relations among words. The similarity measure between two graphs is defined, considering both the vertices and edges. The KNN (K Nearest Neighbors) is modified into the graph based version where data items are classified based on the similarity between graphs, and applied to the text categorization tasks. The adjacency matrix is adopted as representation of each graph in this research.

Let us mention some benefits which are expected from this research. By avoiding the problems from encoding texts into numerical vectors, we expect the better performance of the proposed version than the traditional version of the KNN. Since the graphs are more symbolic text representations than numerical vectors, we expect more transparency in encoding so. We expect more compactness of representing texts than numerical vectors for processing texts more efficiently. Hence, the goal of this research is to implement the text categorization 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 two kinds of non-numerical vector based machine learning algorithms: table based machine learning algorithms and string vector based machine learning algorithms. 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 applying the modernized machine learning algorithms for the text categorization and its related tasks. We mention the word categorization to which the modernized KNN algorithm is applied, as a task which is related with the text categorization. We present the cases of applying the modernized KNN algorithm to the text categorization which

is covered as the challenge of this research. We consider the text clustering where the modernized AHC algorithm is applied as another related task. This section is intended to survey the cases of applying the modernized KNN algorithm and the modernized AHC algorithm, for the text categorization and its related tasks.

Let us mention the previous cases of applying the graph based KNN version to the word categorization. In 2006, Jo initially proposed the modification of the KNN algorithm into its graph based version as an approach to the word categorization [20]. In 2018, the modernized version was compared with the traditional version as the start of observing its better performance in the word categorization [26]. In 2018, the better performance of the modernized version was completely validated in categorizing words in the three text collection [27]. In the above literatures, we observe the previous cases of using the modernized version of the KNN algorithm for the word categorization.

Let us survey cases of applying the KNN algorithm which processes graphs directly as a modernized version for the text categorization which is covered in this research. The proposed version of the KNN algorithm is initially asserted as the approach to the text categorization by Jo in 2018 [28]. Its better performance than the traditional version was discovered in classifying texts in a small text collection in 2019 [33]. This research is aimed to finalize validating the better performance of the version which receives a graph as its input data, in the text classification. In the above literatures, we mention the graph based KNN algorithm which is used as an approach to the text categorization.

Let us explore the previous works where the graph based AHC algorithm is applied for clustering texts. The graph based version was initially asserted as an approach to the text clustering by Jo in 2017 [22]. He started to observe its better performance than the traditional AHC algorithm in a toy experiment, in 2019 [34]. The empirical validation of the better performance was finalized by real experiments in 2020, but not published, yet [36]. The metric which is used for evaluation the clustering algorithms in those experiments was proposed by Jo and Lee in 2007 [6].

We explored the previous cases of applying the proposed version of KNN algorithm to the tasks which are relevant to this research. The text categorization which is covered in this research is aimed to assign topics to texts, depending on their contents. The KNN version which is adopted in this research processes graphs, directly, and it was applied to the word categorization, as well as the text categorization. The AHC algorithm which was applied to the text clustering was modified in the previous works by the same style of doing the KNN algorithm. The goal of this research is to validate completely the better performance of the proposed KNN algorithm as the approach to the text categorization through the real text collections.

B. Encoding Schemes

This section is concerned with the various schemes of encoding texts into structured data. In this research, we propose that texts should be encoded into graphs as structured data. We mention other structured data, such as numerical vectors, tables, and string vectors, in surveying previous works. In the previous works which are explored in this section, we will present the modified versions of the KNN algorithm which process the structured data, directly. This section is intended to survey the previous works on encoding texts into three kinds of structured data.

Let us review the previous cases of encoding words or texts into numerical vectors. In 2018, texts were encoded into numerical vectors, in using the AHC algorithm for clustering texts [29]. In 2019, words were encoded into numerical vectors, in using the KNN algorithms for classifying them [30]. In 2019, texts were encoded so in using the KNN algorithm for classifying them [35]. The similarity between numerical vectors is computed by considering the feature similarities and the feature value similarities, to prevent the poor discriminations among sparse vectors.

Let us survey the previous works where texts are encoded into tables. In 2008, Jo and Cho initially tried to encode texts into tables in the text categorization [13]. In 2008, texts were encoded so and the online clustering algorithm was modified as the approach to the text clustering [9]. In 2015, Jo proposed the table matching algorithm where texts are encoded into tables as the approach to the text categorization [19]. In the above literatures, we presented the previous cases where texts are encoded into tables.

Let us mention the previous cases of encoding a text into a string vector as an ordered finite set of strings. In 2018, texts were encoded into string vectors for modifying the KNN algorithm into the string vector based version as the approach to the text categorization [31]. In 2018, the text summarization is viewed into the classification of each paragraph into summary or non-summary, and the string vector based version of the KNN algorithm is applied to the task [32]. The AHC algorithm is modified as the approach to the text clustering into the version where a text is encoded into a string vector, in 2020 [37]. In the above literatures, we present the cases of encoding texts into string vectors for modifying the KNN algorithm and the AHC algorithm.

We surveyed the previous works on the schemes of encoding texts into structured forms. Texts were encoded into numerical vectors, and the similarity metric which considers the feature similarities was proposed. Texts were encoded into tables and the similarity metric between tables based on their shared entries was proposed. Texts were encoded into string vectors, and the semantic similarity between them was proposed for modifying the KNN algorithm and the AHC algorithm as the approaches to the text mining tasks. In this research, texts are encoded into graphs, and the similarity

metric between two graphs which is described in Section III-B is proposed.

C. Table based Machine Learning Algorithms

This section is concerned with the previous works on the table based approaches to text mining tasks. We will present the classification algorithm and the clustering algorithm which processes tables, instead of numerical vectors. We will mention the table based matching classification algorithm, the table based matching clustering algorithm, and the table based KNN algorithm, as the kind of the non-numerical vector based machine learning algorithms. The significance of the previous works which are surveyed in this section is to try to solve the problems in encoding texts into numerical vectors, such as huge dimensionality, sparse distribution, and the poor transparency. This section is intended to explore the previous works on the three table based algorithms as the approaches to the text mining tasks.

Let us survey the previous works on the table based machine algorithm as an approach to the text categorization. In 2008, Jo and Cho initiated solving the problems in encoding texts into numerical vectors by proposing initially the table based matching algorithm [13]. It was applied to the soft text categorization where each text is allowed to be classified into more than one category, in 2008 [9]. It was improved and stabilized as the approach to the text categorization, in 2015 [19]. In the above literatures, we present the table based matching algorithm for avoiding the problems in encoding texts into numerical vectors.

Let us survey the previous works on the clustering algorithm which processes tables, directly. The table based matching algorithm was initially applied to the text clustering, as well as the text categorization, in 2017 [8] Its performance was validated in the real text collection, 20NewsGroup, in 2008 [14]. The online linear clustering algorithm was modified into the table based version as the approach to the text clustering, in 2008 [10]. In the above literatures, we presented the table based clustering algorithm which clusters tables, instead of numerical vectors.

Let us explore the previous works on the table based KNN algorithm as a non-numerical vector based classifier. It was proposed as the approach to the text categorization by defining the similarity between tables as one between texts, in 2017 [23]. The version of the KNN algorithm was applied to the text summarization which is mapped into an instance of text categorization, in 2017 [24]. It was applied to the text segmentation as one more text categorization instance, in 2017 [25]. In the above literatures, we presented the proposal of the table based KNN algorithm as a non-numerical vector based classification algorithm and its applications to the text categorization instances.

We surveyed the previous works on the table based machine learning algorithms as the approaches to the text mining tasks. The table based matching algorithm was proposed and stabilized as an approach to the text categorization. It was applied to the text clustering, as well as the text categorization, as a clustering algorithm. The KNN algorithm was modified into the table based version which processes tables, directly. In this research, the KNN algorithm was modified into the graph based version which processes graphs, directly, as the alternative one to the table based version.

D. String Vector based Machine Learning Algorithms

This section is concerned with the previous works on the string vector based machine learning algorithms which are the approaches to the text categorization and the text clustering. A string vector is defined as an ordered finite set of strings; numerical values are replaced by strings as the elements in a vector. The SVM with the string vector kernel function, the NTC (Neural Text Categorizer), and the NTSO (Neural Text Self Organizer) will be mentioned the typical string vector based machine learning algorithms in surveying the previous works. They are used for categorizing and clustering texts 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 kernel function which indicates the similarity between two string vectors. The string vector kernel was initially defined and implemented based on the inverted index where each word is linked with texts which include itself, in 2007 [5]. The string vector kernel is implemented by defining the similarity matrix as a square matrix which consists of semantic similarities between words, in advance, in 2007 [7]. The string vector kernel was used for modifying the SVM into its string vector based version as the approach to the text categorization [11]. In the above literatures, the string vector kernel was defined as the similarity between string vectors, and the SVM was modified using it.

Let us explore the previous works on the NTC (Neural Text Categorizer) as a string vector based neural networks. It was initially created and applied to the text categorization by Jo in 2008 [12]. Its better performance was empirically validated in both the hard text categorization and the soft text categorization, in 2010 [15]. The NTC was applied for classifying texts in Arabian by Abainia et al., in 2015 [18], and mentioned as an innovative neural networks by Vega and Mendez-Vazquez, in 2016 [21]. In the above literatures, the proposal, the application, and the citation of the NTC are presented.

Let us survey the previous works on the NTSO (Neural Text Self Organizer) as another string vector based neural networks. It was initially proposed as the approach to the text clustering by Jo and Japkowicz, in 2005 [2]. It was mentioned as an innovative neural networks by Zheng et al. in 2006 [4]. The progress of the research on the NTSO was finalized by its complete validation in the real experiments

on the text clustering, in 2010 [16]. In the above literatures, we presented the initial proposal and the complete validation of the NTSO.

In this research, texts are encoded into graphs, instead of string vectors. In the above literatures, text are encoded into string vectors as another way of avoiding the problems in encoding texts into numerical vectors. It takes very much time for building the similarity matrix from a corpus as the basis for computing the semantic operations on string vectors. The semantic similarities among words depends strongly on the corpus; the semantic similarity between two words may be different, depending on the corpus. It is necessary to define and characterize mathematically more semantic operations for modifying other machine learning algorithms into their string vector based versions.

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 categorization, 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 vector based KNN version as the approach to the text categorization. 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. The graph is defined in the context of the data structure as the two sets: the vertex set and the edge set. The words in the graph which represents a text are given as vertices. A semantic similarity between words is given as an edge, and computed based on collocations of words in a corpus. This section is intended to describe the steps of encoding a text into a graph, in detail.

The process of indexing a text into a list of words as vertices is illustrated in Figure 1. In representing a text into a graph, words are defined as vertices. A single text is given as the input in the left side in Figure 1, and N words are given as the results from indexing the text in the right side. The basic steps of indexing a text for generating vertices are the tokenization, the stemming, and the stopword removal. The vertex set is constructed in this step for constructing a graph from the input text.

The definition of edges in the graph which represents a text is illustrated in Figure 2. The N words were already generated by the process which is illustrated in Figure 1. All possible pairs are generated from the N words and the semantic similarity is computed 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. We need only some edges with higher similarities, instead of all complete edges for building a graph.

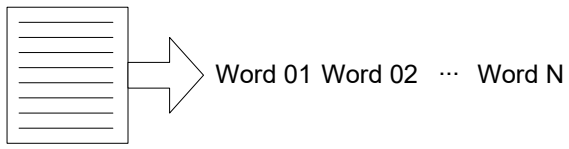

Figure 1. Vertex Definition

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

Figure 2. Edge Definition

The graph which represents a text is illustrated in Figure 3, as a simple example. The four words, information, computer, business, and system, are given as the vertices of the graph. The edges in Figure 3 are given as the complete edges, and the weight of each edge indicates the similarity between words as vertices. The similarity between vertices becomes an edge identifier of the graph. A corpus is needed for computing the similarity between words, based on their collocations.

Let us make some remarks on the process of encoding

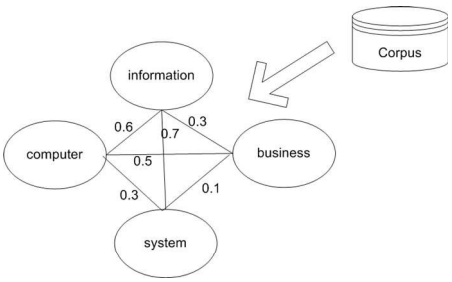

Figure 3. Graph representing a Text

a text into a graph. The graph is defined formally as the two sets: the vertex set and the edge set. In the graph which represents a text, its vertices are given as words, and its edges are given as the similarities among words. The similarity which weights each edge is computed based on the collocations of words in texts. In this research, each graph is represented into an edge set, in the implementation level.

B. Similarity Metric

This section is concerned with the computation of similarity between graphs. A graph is represented into a set of edges in the implementation level. The similarity between edges is computed and it is expanded into one between two graphs. The similarity between two graphs is always given as a normalized value between zero and one, and proportional to the shared edges between two graphs. This section is intended to describe the similarity metric between two graphs which is proposed in this research.

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),

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

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),

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

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),

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

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.

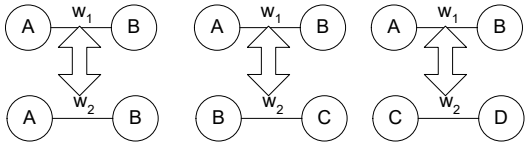

Figure 4. Three Cases in computing Edge Similarity

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 two graphs which is covered in this section. The similarity between two edges is computed, considering the three cases. The maximum of the similarities of an edge with ones in a graph is the similarity between an edge and a graph. Average over the similarities of edges of first graph with ones in the second graph becomes the similarity between two graphs. The similarity metric between two graphs is utilized for modifying the KNN algorithm into the graph based version which processes graphs directly.

C. Proposed Version of KNN

This section is concerned with the graph based KNN algorithm as the approach to the text classification. In the previous section, we described the similarity metric between two graphs which is used for modifying the KNN algorithm into the proposed version. A novice text is encoded into a graph, and its similarities with the training graphs are computed, using the similarity metric. Like the traditional version of the KNN, the labels of nearest neighbors are voted for deciding one of the novice one. This section is intended to describe the modified version of the KNN algorithm, as an approach to the text classification.

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.

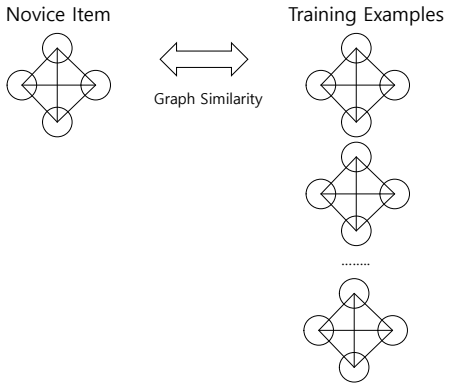

Figure 5. Similarities of a Novice Graph with Sample Ones

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}.

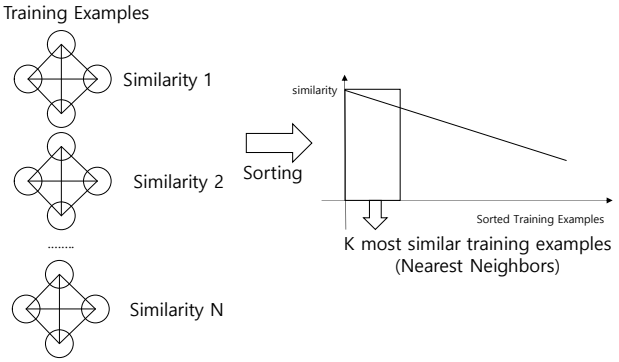

Figure 6. Selection of Nearest Neighbors from Training Examples

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),

$$C_{\max} = \underset{i=1}{\overset{|C|}{\arg\max}} Count(C_i, Ne) \quad (12)$$

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.

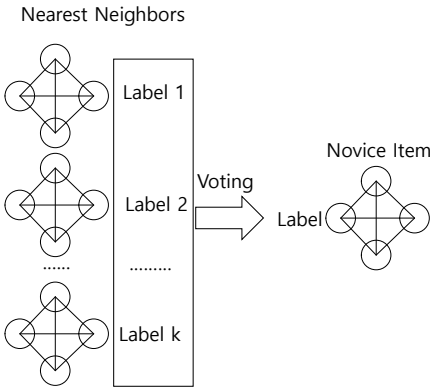

Figure 7. Voting Labels of Training Examples for deciding One of Novice Example

Let us make some remarks on the graph based KNN algorithm as the approach to the text categorization. In using the version, it is assumed that the sample texts and a novice text are encoded into graphs. The similarities of a novice graph with the sample graphs is computed by the similarity metric which is described in Section III-B. The sample graphs are ranked by their similarities with the novice graph, and the K sample graphs with their highest similarities are selected as its nearest neighbors. The labels of the nearest neighbors are voted for deciding the label of the novice one.

D. Text Categorization System

This section is concerned with the system architecture and the execution flow of the text categorization system. The KNN algorithm which processes graphs directly is adopted as the approach to the text categorization, and was already described in Section III-C. In this system, a novice text is encoded into a graph, and classified by the KNN algorithm. We present the system architecture and the execution process in the step of designing the system, and consider its implementation in Java or Python in the next research. This section is intended to describe the sampling process, the system architecture, and the execution process of the system.

In Figure 8, gathering texts as samples for each topic is illustrated. The M topics are predefined as a list under the assumption which the text categorization belong to the flat classification. Texts are gathered and allocated to each topic, and encoded into graphs by the process which was described in Section III-A. The M groups of graphs are given as the training set in the system, as shown in the bottom of Figure 8. The hierarchical text categorization where the categories are predefined as a tree will be considered in the next research.

The system architecture of the text classification system is illustrated in Figure 9. In the encoding module, texts are encoded into graphs by the process which is described in Section III-A. The role of the similarity computation module is to compute the similarities of a graph which represents a novice text and with ones which represent the sample texts, and selecting some with their highest similarities as the nearest neighbors. The role of the voting module is to decide he label of the novice text by voting ones of the nearest neighbors. The role of the proposed system is to classify unlabeled texts.

The execution process of the text classification system is illustrated in Figure 10. The similarity matrix is constructed for defining edges from the sample texts, and both the sample texts and a novice one are encoded into graphs. The similarity between graphs is computed as one between a novice and a sample in the execution of the KNN algorithm. The category of the novice one is decided by voting ones of the nearest neighbors. The category of the novice item is generated as the output of the system.

Let us some remarks on the system architecture and the execution process of the text clustering system which are presented in Figure 9 and 10. Encoding of texts into graphs and the similarity between them are proposed in this research. The KNN algorithm is modified into the graph based version by defining the similarity between graphs as one between a novice text and a sample text. This research provides the system architecture and the execution flow which are needed for doing the general design. In the next research, we consider the detail design and the source code

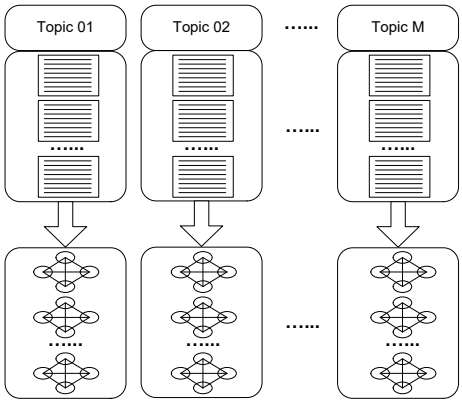

Figure 8. Collection of Sample Texts

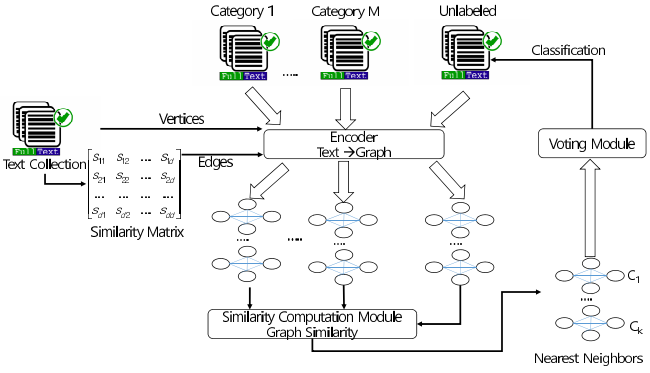

Figure 9. System Architecture

for implementing 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 categorization on the collection, NewsPage.com. In Section

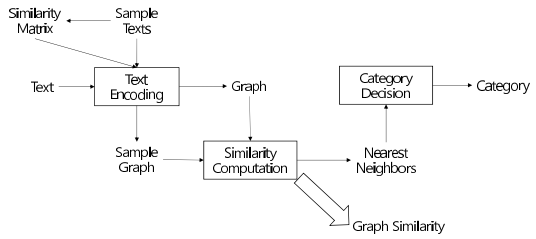

Figure 10. Execution Process

IV-B, we show the results from applying it for categorizing texts 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 categorizing texts 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. The four categories are predefined in this collection, and texts are gathered from the collection category by category as labeled ones. Each text is classified exclusively into one of the four categories. In this set of experiments, we apply the traditional and proposed version of KNN to the classification task, without decomposing it into the binary classifications, and use the accuracy as the evaluation measure. Therefore, in this section, we observe the performance of the both versions of KNN by changing the input size.

In Table I, we specify the text collection, NewsPage.com, which is used in this set of experiments. This text collection was used for evaluating approaches to text categorization in previous works [19]. In the collection, the four categories are predefined: Business, Health, Internet, and Sports, and 375 texts are selected at random in each category. In each category, the set of 375 texts is partitioned into the 300 texts as training ones and the 75 texts as test ones. The text collection was built by copying and pasting individual news articles from the web site, newspage.com, in 2005, as plain text files whose extension is 'txt'.

Table I
THE NUMBER OF TEXTS IN NEWSPAGE.COM

Category	#Texts	#Training Texts	#Test Texts
Business	500	300	75
Health	500	300	75
Internet	500	300	75
Sports	500	300	75
Total	2000	1200	300

Let us mention the experimental process for validating empirically the proposed approach to the task of text categorization. In this collection, the texts are labeled with one of the four categories which are presented in Table I, and they are encoded into numerical vectors and graphs. For each test example, the KNN computes its similarities with the 1200 training examples and selects the three most similarity training examples as its nearest neighbors. Each of the 300 test examples is classified into one of the four categories: Business, Sports, Internet, and Health, by voting the labels of its nearest neighbors. 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 of KNN algorithm.

In Figure 11, we illustrate the experimental results from categorizing texts, 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. In the x-axis, each group indicates the input size which is the dimension of numerical vectors which represent texts. In each group, the gray bar and the black bar indicate the achievements of the traditional version and the proposed version of KNN algorithm, respectively. In the x-axis, the most right group indicates the average over the accuracies of the left groups.

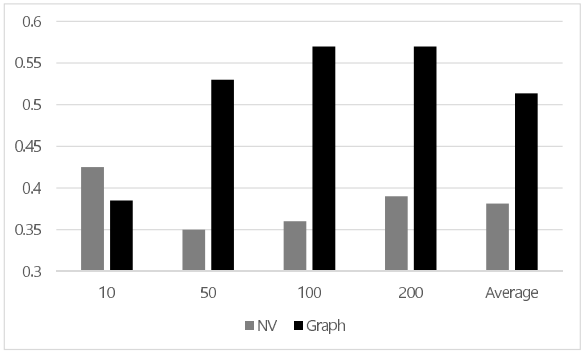

Figure 11. Results from Classifying Texts in Text Collection: NewsPage.com

Let us make the discussions on the results from doing the text categorization using the both versions of KNN algorithm, as shown in Figure 11. The accuracy which is the performance measure of the classification task is in the range between 0.35 and 0.52. The proposed version of KNN algorithm works strongly better in the three input sizes, 50, 100, and 200. It loses in the input size, 10. From this set of experiments, in spite of the fact, we conclude that the proposed version works strongly better than the 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. The three categories are predefined in the collection, and labeled texts are prepared from it. Each

text is classified exclusively into one of the three categories. We do not decompose the given classification into binary classifications and use the accuracy as the evaluation measure. Therefore, in this section, we observe the performances of the both versions of KNN algorithm with the different input sizes.

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 approaches to text categorization. The three categories, 'Car', 'Electronics', and 'Hotel', are predefined, and all texts are used for evaluating the approaches to text categorization, in this set of experiments. We use six texts in each category among all texts as the test set as shown in Table II. We obtained the collection by downloading it from the web site, http://archive.ics.uci.edu/ml/machine-learning-databases/opinion/.

Table II
THE NUMBER OF TEXTS IN OPINIOPSIS

Category	#Texts	#Training Texts	#Test Texts
Car	23	17	6
Electronic	16	10	6
Hotel	12	6	6
Total	51	33	18

We perform this set of experiments by the process which is described in Section IV-A. We use all of 51 texts which are labeled with one of the three categories and encode them into numerical vectors and graphs with the input sizes: 10, 50, 100, and 200. For each test example, the both versions of KNN computes its similarities with the 33 training examples and select the three most similar training examples as its nearest neighbors. Each of the 18 test examples is classified into one of the three categories, by voting the labels of its nearest neighbors. 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 categorizing texts 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 both versions by an input size. In each group, the gray bar and the black bar indicate the achievements of the traditional version and the proposed version of KNN algorithm, respectively. In Figure 12, the most right group indicates the averages over results over the left four groups. Therefore, Figure 12 presents the results from classifying each text into one of the three categories by the both versions, on the text collection, Opinosis.

We discuss the results from doing the text categorization using the both versions of KNN algorithm, on Opinosis, shown in Figure 12. The accuracy values of the bother versions range between 0.55 and 1.0. The proposed version

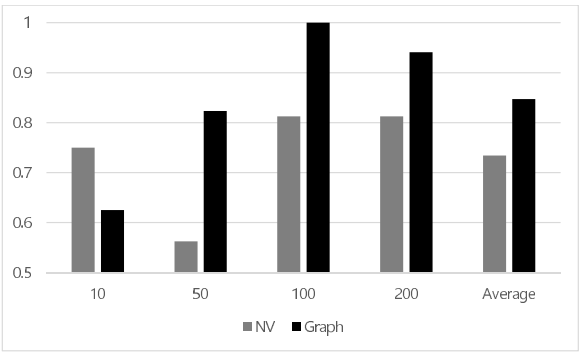

Figure 12. Results from Classifying Texts in Text Collection: Opiniopsis

works better than the traditional one in the three input sizes: 50, 100, and 200. It shows the perfect results in the input size: 100. From this set of experiments, we conclude that the proposed version works better than the traditional one, in averaging the four 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 the text collection, 20NewsGroup I. In this set of experiments, we predefine the four general categories in this collection, and gather texts from it category by category as the classified ones. Each text is classified exclusively into one of the four categories. We apply the KNN algorithms directly to the given task without decomposing it into binary classifications, and use the accuracy as the evaluation measure. Therefore, in this section, we observe the performances of the both versions with the different input sizes.

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 select 375 texts from 4000 or 5000 texts at random. The 375 texts is partitioned into the 300 texts in the training set and the 75 texts in the test sets, as shown in Table III. We obtain the collection, 20NewsGroup, by downloading from the web site, https://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html, as one of the standard text collection for evaluating approaches to text categorization.

The experimental process is identical is that in the previous sets of experiments. In each category, we select the 375 texts at random and encode them into numerical vectors and graphs with the input sizes, 10, 50, 100, and 200. For each test example, we compute its similarities with the 1200 training examples, and select the three similar ones as its nearest neighbors. The versions of KNN algorithm classify

Table III
THE NUMBER OF TEXTS IN 20NEWSGROUPS I

Category	#Texts	#Training Texts	#Test Texts
Comp	5000	300	75
Rec	4000	300	75
Sci	4000	300	75
Talk	4000	300	75
Total	17000	1200	300

each of 300 test examples into one of the four categories: comp, rec, sci, and talk, by voting the labels of its nearest neighbors. We also use the classification accuracy as the evaluation measure in this set of experiments.

In Figure 13, we illustrate the experimental results from classifying the texts into one of the four topics 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. Figure 13 presents the results from classifying each text into one of the four broad categories. In this set of experiments, note that the task is not decomposed into binary classifications.

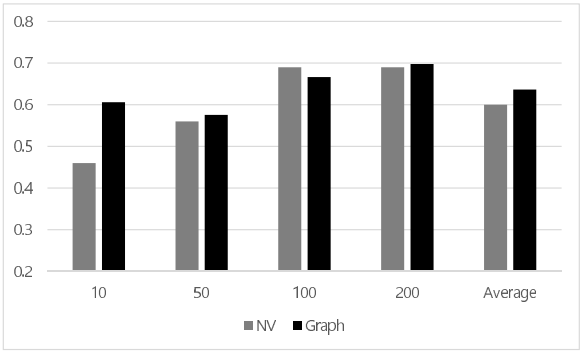

Figure 13. Results from Classifying Texts in Text Collection: 20NewsGroup I

Let us discuss the results from classifying the texts using the both versions of KNN algorithm on the broad version of 20NewsGroups into one of the four categories, as shown in Figure 3. The accuracies of the both versions range between 0.45 and 0.7. The proposed version shows its better performance in two of the four input sizes. It keeps its competitiveness with the traditional one in the others. From this set of experiments, we conclude that the proposed version wins over the traditional over, in averaging the achievements of the four input sizes.

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. In this set of experiments, the four specific categories are predefined in this collection. Each text is exclusively classified into one of the four categories, like the previous sets of experiments. We apply the two versions of KNN algorithm, directly to the classification task, without decomposing it into binary classifications, and use the accuracy as the evaluation metric. Therefore, in this section, we observe the performances of the both versions of KNN algorithm with the different input sizes.

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 category, we select 375 texts among approximately 1000 texts, at random. In each category, the set of 375 texts is partitioned into the training set of 300 texts and the test set of 75 texts, like the case in the previous set of experiments. The task in the set of experiments in Section IV-C is a broad classification, whereas that in this set of experiments is a specific classification.

Table IV
THE NUMBER OF TEXTS IN 20NEWSGROUPS II

Category	#Texts	#Training Texts	#Test Texts
Electro	1000	300	75
Medicine	1000	300	75
Script	1000	300	75
Space	1000	300	75
Total	4000	1200	300

The process of doing this set of experiments is same to that in the previous sets of experiments. We select the balanced number of texts from the collection over categories, and encode them into the representations with the input sizes which are identical to those in the previous set of experiments. We use the two versions of KNN algorithm for their comparisons. Using the two versions of KNN algorithm, we classify each text in the test set into one of the four specific categories within the general category, 'sci': 'electro', 'medicine', 'script', and 'space'. We use the accuracy as the evaluation metric, like the previous set of experiments.

We present the experimental results from classifying the texts 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. The texts are classified directly to one of the four categories like the cases in the previous sets of experiments.

Let us discuss on the results from classifying the texts on the specific version of 20NewsGroups, as shown in Figure 14. The accuracies of the both versions range between 0.4 and 0.8. The proposed version shows its better performance

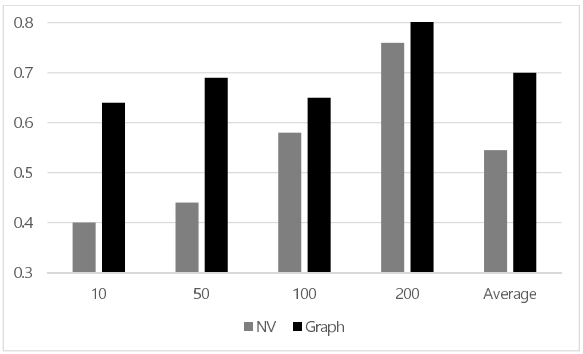

Figure 14. Results from Classifying Texts in Text Collection: 20News-Group II

in all of the four input sizes. The performance of both versions is correlated with the input size, as shown in Figure 14. From this set of experiments, it is concluded that the proposed version have its outstandingly better performance, by averaging over the accuracies of the four input sizes.

V. Conclusion

Let us discuss the entire results from classifying texts using the two versions of KNN algorithm. The both versions is compared with each other in the task of text categorization, in these sets of experiments. The proposed version show its better results in all of the four collections. The accuracies of the traditional version range between 0.35 and 0.81, while those of the proposed version range between 0.49 and 1.0. From the four sets of experiments, we conclude that the proposed version improves the text categorization 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 classifying 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 categorization 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 and N. Japkowicz, "Text Clustering using NTSO", 558-563, The Proceedings of IJCNN, 2005.

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

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

[5] 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.

[6] T. Jo and M. Lee, "The Evaluation Measure of Text Clustering for the Variable Number of Clusters", 871-879, Lecture Notes in Computer Science, Vol 4492, 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] K. Yi, T. Jo, M. Lee, and Y. Choi, "Modifying Online Text Clustering Algorithm using Inverted Index based Operation", 150-153, The 2007 International Conference on Semantic Web and Web Services, 2007.

[9] 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.

[10] 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.

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

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

[13] 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.

[14] T. Jo and G. Jo, "Table based Matching Algorithm for Clustering Electronic Documents in 20NewsGroups", 66-71, IEEE International Workshop on Semantic Computing and Applications, 2008.

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

[16] 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.

[17] D. Allemang and J. Hendler, "Semantic Web for the Working Ontologies", Mrgan Kaufmann, 2011.

[18] 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.

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

[20] 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.

[21] 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.

[22] T. Jo, "Graph based AHC Algorithm for Text Clustering", 309-314, The Proceedings of Computer Science and Computational Intelligence, 2017.

[23] T. Jo, "Table based KNN for Article Classification", 271-276, The Proceedings of 19th International Conference on Artificial Intelligence, 2017.

[24] T. Jo, "Table based KNN for Text Summarization", 31-36, The Proceedings of 4th International Conference on Advances in Big Data Analysis, 2017.

[25] T. Jo, "Content based Segmentation of Texts using Table based KNN", 27-32, The Proceedings of 16th International Conference on Advances in Information and Knowledge Engineering, 2017.

[26] 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.

[27] 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.

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

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

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

[31] 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.

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

[33] 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.

[34] T. Jo, "Graph based Version for Clustering Texts in Current Affair Domain", 171-174, The Proceedings of 15st International Conference on Data Science, 2019.

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

[36] T. Jo, "K Nearest Neighbor specialized for Word Categorization in Current Affairs by Graph based Version", 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.