Table based K Nearest Neighbor for Text Classification

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

***Abstract*—This article proposes the modified KNN (K Nearest Neighbor) algorithm which receives a table as its input data and is applied to the text categorization. The motivations of this research are the successful results from applying the table based algorithms to the text categorizations in previous works and the expectation of synergy effect between the text categorization and the word categorization. In this research, we define the similarity metric between two tables representing texts, 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 using the table based KNN algorithm, it is easier to trace results from categorizing texts.**

I. Introduction

The task of text categorization refers to the process of classifying each text into its own topic or category, as an instance of pattern classification. Even if other kinds of approaches are available, assuming that the supervised machine learning algorithms are used as the main approaches, we need to predefine a finite list of topics or categories and allocate sample texts to each topic or category as the preliminary tasks. Afterward, by learning sample labeled texts, we build the classification capacity given as symbolic rules, equations, and/or parameters of statistical models, depending on the type of machine learning algorithms. According to the constructed classification capacity, the novice texts which are given separately from the sample ones are classified. Although there are various types of text categorizations such as hard text categorization, soft text categorization, and hierarchical text categorization, the scope of this research is restricted to only hard text categorization.

This research is motivated by the three addenda. First, encoding texts into numerical vectors for using the traditional approaches leads to the three problems: huge dimensionality, sparse distribution, and poor transparency [3]. Second, although the table based approach called table matching algorithm was previously proposed as the first approach where texts are encoded into tables, it was very sensitive to noisy texts [3]. Third, previously, we tried to encode texts into string vectors as alternative representations, but we need to define more mathematical definitions, in order to modify and create string vector based versions of machine learning algorithms [14]. Hence, motivated by the three agenda, this research attempts to modify the KNN (K Nearest Neighbor) into the table based version.

Let us mention what we propose in this research. In this research, texts are encoded into tables instead of numerical vectors, in order to avoid the three main problems. We define the similarity measure between two tables which is always given as a normalized value and apply it to the modification of the KNN. The modified version will be used as the approach to the text categorization. Each table which represents a table consists of entries of words and their weights.

We may expect mainly the three benefits from this research. First, we expect more stable performance than the approach used in [3] by avoiding the impact by the text lengths and their noises. Second, by avoiding problems in encoding texts into numerical vectors for using any of traditional machine learning algorithms, the better performance is expected than the traditional version of K Nearest neighbors. The table as the text surrogate provides more transparence where we can guess the content of texts by seeing the surrogate, since it is more symbolic representation than numerical vectors. However, the table size is given as the external parameter of the proposed system, and it impacts the trade-off between the classification reliability and the speed of computing the similarity between two tables.

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 previous works on table based machine learning algorithms and string vector based machine learning algorithms, respectively. Therefore, in this section, we provide the history about this research, by surveying the relevant previous works.

A. Related Tasks

This section is concerned with the previous cases of applying the modernized KNN and AHC to the text categorization and its related tasks. We will mention the word categorization which classifies each word based on its meaning as the task which is related with the text categorization. We will survey the previous cases of applying the modernized KNN algorithm for the text categorization which is covered in this research. We consider the cases of applying the modernized AHC algorithm as well as the modernized KNN algorithm for the text clustering as another related task. This section is intended to survey the cases of applying both modernized machine learning algorithms for the text categorization and the related tasks.

Let us survey the cases of applying the modernized KNN algorithm for the word categorization as a relevant task to the text categorization. In 2016, Jo initially proposed the idea of modifying the KNN algorithm into the table based version as an approach to the word categorization [17]. In 2018, the table based version was compared with the traditional version in the word categorization, observing its better classification performance [25]. In 2018, the KNN algorithm which considers the feature similarities as the alternative version to the table based one was applied to the word categorization [26]. The better performance of the table based version was validated in the word categorization, given as an unpublished paper [37].

Let us explore the cases of applying the table based version of the KNN algorithm for the text categorization which is covered in this research. It was initially mentioned as the approach to the text categorization, in 2017 [20]. In 2018, Jo tried to compare the proposed version with the traditional version, in classifying texts into one among the predefined categories in a small text collection [27]. This research is intended to complete validating the better performance of the table based version than the traditional version in the text classification in real text collections. In the above literatures, we mentioned the application of the modernized version of the KNN algorithm which processes tables directly to the text categorization.

Let us mention the previous cases of applying the table based AHC algorithm for the text clustering as well as the table based KNN algorithm. It was initially asserted that the table based version of the AHC algorithm should be used for clustering texts, by Jo in 2017 [21]. The table base version was compared with the traditional AHC algorithm, and its better performance is discovered in clustering texts in a small text collection [34]. The empirical validation of the better performance was recently complemented in real text collections, but it is not published, yet [38]. The metric for evaluating clustering results was proposed by Jo and Lee in 2007, and called clustering index [5].

We surveyed the previous cases of applying the proposed version of the KNN algorithm for the tasks which are relevant to this research. The text categorization which is covered in this research is the process of assigning a topic or topics to each text among the predefined ones. The proposed version of the KNN algorithm which is used as the approach to the text categorization processes table directly. In the previous works, the proposed version was applied to the word categorization, as well as the text categorization, and the modernized version of the AHC algorithm which was modified in the style of doing the KNN algorithm was applied to the text clustering which is related with the text categorization. The goal of this research is to validate completely the better performance of the table based version through several sets of experiments.

B. Encoding Schemes

This section is concerned with the previous works on various schemes of encoding texts into structured data. In this research, we propose that texts should be encoded into tables as structured data. In this survey, we mention the numerical vectors, the string vectors, and the graphs as alternative structured data to the table. In the previous works, the KNN algorithm is modified into the version which processes such kinds of structured data as the approaches to text mining tasks. This section is intended to survey the previous works on the schemes of encoding texts into structured data.

Let us review the previous cases of encoding the texts into numerical vectors in applying the modernized machine learning algorithms. In 2018, texts were encoded into numerical vectors for using the modernized version of the AHC algorithm, and its better performance was validated empirically in the text clustering [28]. In 2019, the words were encoded into numerical vectors for applying the KNN algorithm as the approach to the word categorization, and its better performance than the traditional version was validated [29]. In 2019, texts were encoded so in using the modernized KNN algorithm for the text classification [35]. In the above literatures, both the KNN algorithm and the AHC algorithm were modernized by considering both the feature similarities and the feature value similarities, in computing the similarity between numerical vectors.

Let us survey the previous works where a text is encoded into a string vector which is a finite ordered set of strings. In 2018, texts were into string vectors for modifying the KNN algorithm as the approach to the text categorization [30]. In 2018, the modified KNN algorithm where texts are encoded so was applied to the text summarization [31]. In 2020, texts were encoded so in modifying the AHC algorithm as the approach to the text clustering [39]. In the above literatures, we present the cases of encoding texts into string vectors for modifying the KNN algorithm and the AHC algorithm.

Let us explore the previous works where words or texts are encoded into graphs in text mining tasks. In 2016, in

order to optimize index, words are encoded into graphs for classifying them into the categories; expansion, inclusion, and removal [18]. In 2018, words were encoded so for modifying the KNN algorithm which was the approach to the word categorization [32]. In 2019, texts were encoded so for modifying the AHC algorithm as the approach to the text clustering [36]. In the above literatures, we presented the cases of encoding words or texts into graphs in the text mining tasks.

We surveyed the cases of encoding texts into other types of structured data. Even if texts are encoded into numerical vectors as the traditional form, the poor discriminations among sparse vectors are prevented by defining the similarity matric which considers the feature similarities. A text was encoded into a string vector which is an ordered finite set of strings; a string vector is one where numerical values are replaced by strings. It may be encoded into a graph where its vertices are given as words and its edges are given as semantic relations among them. In this research, a text is encoded into a table, and the similarity between them will be defined for modifying the KNN algorithm.

C. Table based Machine Learning Algorithms

This section is concerned with the machine learning algorithms which process tables, instead of numerical vectors. The table based machine learning algorithms were proposed in the previous works, in order to solve the problems in encoding texts into numerical vectors. In the previous works, we mention the table based matching classification, the table based matching clustering algorithm, and the table based KNN algorithm, as the typical ones. The similarity matric which is described in Section III-B is used for computing the similarity between tables. This section is intended to survey the previous works on the three table based machine learning algorithms, rather than to describe each of them.

Let us mention the previous works in the table based matching algorithm which processes tables directly, as the table based classification algorithm. It was initially proposed as the approach to the text categorization by Jo and Cho in 2007 [3]. It was applied to the soft text categorization where each text is allowed to be classified into more than one category by Jo in 2008 [8]. It was improved and stabilized as the approach to both the hard text categorization and the soft text categorization in 2015 [16]. In the above literatures, we present the previous research on the table based matching algorithm as the trials of avoiding the problems in encoding texts into numerical vectors by encoding them into other structured data.

Let us survey the previous works on the table based matching clustering algorithm as the approach to the text clustering. It was initially applied to the text clustering in 2007 [7]. The toy experiment which was performed in [7] was expanded into real experiments for validating the clustering performance of the table based matching algorithm [12]. The online linear clustering algorithm was modified into the table based version which clusters tables instead of numerical vectors, in 2008 [9]. In the above literatures, we presented the clustering algorithm which processes tables, instead of numerical vectors.

Let us mention the previous works on the table based KNN algorithm which classified tables directly as a non-numerical vector based machine learning algorithm. In 2017, the KNN algorithm was modified into the table based version as the approach to the text categorization [22]. The modified version of the KNN algorithm which is mentioned above was applied to the text summarization, in 2017 [23]. It was applied to one more task, the text segmentation, in 2017 [24]. In the previous works, we mention the table based KNN algorithm which was applied to text mining tasks, as a non-numerical vector based machine learning algorithm.

We surveyed the previous works on the table based machine learning algorithms as the trial of solving the problems in encoding texts into numerical vectors. The table based matching algorithm was proposed as the approach to the text categorization, and shown its better results than the main approaches to the text categorization, such as the KNN algorithm, the Na?ve Bayes, and the SVM (Support Vector Machine). The table based matching algorithm was applied to text clustering, as well as the text categorization. The KNN algorithm was modified into the table based version, as the approach to the text categorization and the text summarization. As the goal of this research, the research on the table based KNN algorithm which is the approach to the text categorization is finalized by validating its better performance completely in the real text collections.

D. String Vector based Machine Learning Algorithms

This section is concerned with the previous works on the string vector based machine learning algorithms as another kinds of non-numerical vector based ones. The machine learning algorithm which is proposed in this research deals with tables as another kind of non-numerical vectors. The previous works which are surveyed in this section are about the machine learning algorithms which process string vectors directly, and the string vector is defined as an ordered finite set of strings. The significance of the previous works is to provide another kind of solution to 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.

Let us survey the previous works on the string vector kernel for modifying the SVM. In 2007, the similarity between string vectors was defined as the string vector kernel, and implemented based on the inverted index of words [4]. In 2007, the string vector kernel was implemented by building the similarity matrix as the alternative way [6]. In 2008, the SVM was modified into its string vector

based version, and its better performance than the KNN and the Naive Bayes, and the traditional version of SVM, was presented in the text categorization [10]. In the above literatures, the string vector kernel function was defined and used for modifying the SVM as the approach to the text categorization.

Let us explore the previous works on the NTC (Neural Text Categorizer) as a string vector based neural networks. It was created by Jo in 2008 as the approach to the text categorization [11]. Its better performance than the KNN, the SVM, and the Naive Bayes was empirically validated in both the hard text categorization and the soft one, in 2010 [13]. It was applied for classifying Arabic texts by Abainia et al., in 2015 [15], and it was mentioned as an innovative neural networks by Vega and Mendez-Vazquez, in 2016 [19]. In the above literatures, it is presented that the NTC was proposed, used to the text categorization, and cited in other literatures.

Let us review the previous works on the NTSO (Neural Text Self Organizer) as another string vector based neural networks. The NTSO was initially proposed as the approach to the text clustering by Jo and Japkowicz, in 2005 [1]. It was mentioned as an innovative neural networks in that it processes directly string vectors, instead of numerical vectors, by Zheng et al. in 2006 [2]. Its better clustering performance than the k means algorithm and the Kohonen Networks was completely validated in the real experiments on the text clustering in 2010 [14]. In the above literatures, we present the initial proposal, the citation, and the empirical validation of the NTSO as the approach to the text clustering.

Texts are encoded into tables as alternative structured data to the string vectors, in this research. In the above literatures, they are encoded into string vectors for using one of the machine leaning algorithms. It took very much time for building the big sized similarity matrix which is basis for performing the operations on string vectors from the corpus. The similarity matrix which defines the semantic similarities between strings is strongly dependent on the corpus. We need to define and characterize mathematically more semantic operations on strings for modifying more machine learning algorithms into their string vector based versions.

III. Proposed Approach

This section is concerned with the table based KNN (K Nearest Neighbor) as the approach to text categorization, and it consists of the three sections. In Section III-A, we describe the process of encoding a text into a table. In Section III-B, we do formally that of computing a similarity between tables into a normalized value between zero and one. In Section III-C, we mention the proposed version of KNN together with its traditional version. In Section III-D, we present the 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 table. The table which represents a text is viewed as a list of entries, and each entry consists of a word and its weight which indicates its importance degree in the text. The table is constructed from a text with the three steps: text indexing, word weighting, and table size optimization, and the detail explanation about each step will be provided, subsequently. A table is modeled as a set of entries, and used for computing a similarity between tables. This section is intended to describe each step of representing a text into a table, in detail.

The process of indexing a text into a list of words is illustrated in Figure 1. A single text is given as the input. The text is mapped into a list of words by the indexing process. The tokenization, the steaming, and the stopword removal are the basic steps of the text indexing. The steps are explained in detail in [33].

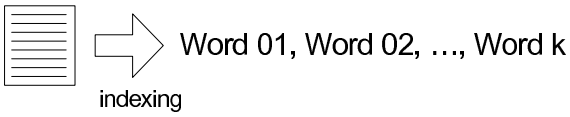

Figure 1. Text Indexing

The second step where weights which are computed by an equation are assigned to words in the table is illustrated in Figure 2. A list of words is gathered from a text in the previous step as shown in the left side in Figure 2. The equation for computing the TF-IDF (Term Frequency Inverse Document Frequency) weight is presented in the bottom of Figure 2, and the word weights are computed in the table by the equation. In the table, each row corresponds to an entry and the two columns in each entry correspond to a word and its TF-IDF weight. When the corpus is not available, the frequency or the relative frequency may be used, instead of the TF-IDF weight.

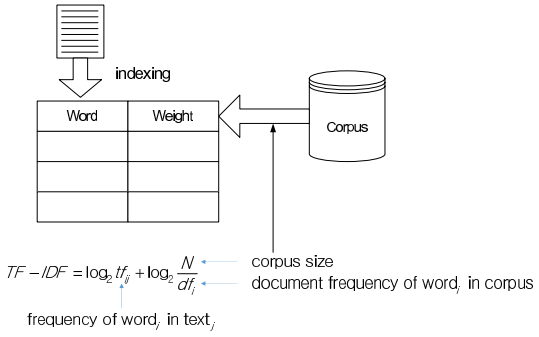

Figure 2. Word Weighting

The process of downsizing the table for more efficient processing is illustrated in Figure 3. Because it takes the quadratic complexity for processing tables to their sizes, it

is important to cut the table size enough for the reliability. The entries in the table are sorted by their weights, and ones with their higher weights are selected. Too much downsize of the table is the cause of frequent zero values which happen in computing similarities among tables. If a short text is encoded into a too small sized table, we need to add more entries from external sources.

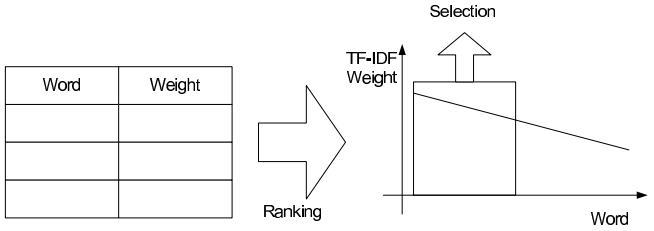

Figure 3. Table Downsizing

Let us make some remarks on the process of encoding texts into tables. The table which represents a text consists of entries, each of which consists of a word and its weight. There are several schemes of weighting a word in each entry: the relative frequency of the word in the text and the TF-IDF weight. Because it takes the high computation complexity to the table size, it should be minimized, maintaining the reliable computation. We need to define more advanced operations on tables, for modifying more advanced machine learning algorithms in this style.

B. Similarity between Two Tables

This section is concerned with the proposed metric for computing a similarity between two tables. The metric which is covered in this section is defined as a binary operation on tables for implementing the machine learning algorithm which process them directly. A function which maps a table into a word set is defined, and the similarity between tables is computed based on their shared entries. The similarity between two tables is always given as a normalized value, and proportional to shared entries. This section is intended to describe the process of computing the similarity between tables.

The function of a table for mapping it into a set of words is illustrated in Figure 4. The table is expressed into a set of entries, each of which consists of a word and its weight, as shown in Equation (1),

$$\begin{aligned} T = \{&(word_1, weight_1), (word_2, weight_2), \\ &\ldots, (word_{|T|}, weight_{|T|})\} \end{aligned} \tag{1}$$

The function, F, of the table, T is defined for taking a set of words as shown in equation (2),

$$F(T) = \{word_1, word_2, \ldots, word_{|T|}\} \tag{2}$$

The table is converted into a bag of words as the role of the function, F, The function, F, is used for generating a table of its entries which are shared by two tables.

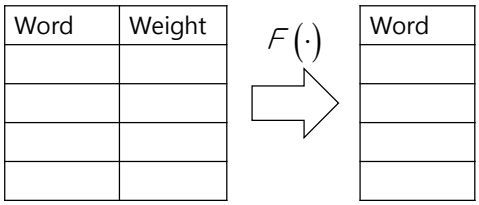

Figure 4. Mapping Table into Word Set

Let us mention the process of computing the similarity between two tables which represent texts. The two tables are expressed as follows:

$$\begin{aligned} T_1 = \{&(word_{11}, weight_{11}), (word_{12}, weight_{12}), \\ &\ldots, (word_{1|T_1|}, weight_{1|T_1|})\} \\ T_2 = \{&(word_{21}, weight_{21}), (word_{22}, weight_{22}), \\ &\ldots, (word_{2|T_2|}, weight_{2|T_2|})\} \end{aligned}$$

The two tables are mapped into sets of words by applying the function, F, as follows:

$$\begin{aligned} F(T_1) &= \{word_{11}, word_{12}, \ldots, word_{1|T_1|})\} \\ F(T_2) &= \{word_{21}, word_{22}, \ldots, word_{2|T_1|})\} \end{aligned}$$

and the set of shared words is obtained by applying the

intersection the two sets as shown in equation (3),

$$F(T_1) \cap F(T_2) = \{sword_1, sword_2, \ldots, sword_k\} \quad (3)$$

The shared table is constructed by taking their weights from the two table, T_1 and T_2, as follows:

$$ST = \{(sword_1, sweight_{11}, sweight_{21}), (sword_1, sweight_{12}, sweight_{22}), \ldots, (sword_k, sweight_{1k}, sweight_{2k})\}$$

For each shared word, $sword_i$, $sweight_{1i}$ is the weight from the table, T_1, and $sweight_{2i}$ the weight from the table, T_2.

Let us mention the process of computing the similarity between two tables, based on the shared table. It consists of the entries, each of which has the three components: a word, and its dual weights from the two input tables. The similarity between the two tables, T_1 and T_2, is computed by equation (4),

$$sim(T_1, T_2) = \frac{\sum_{i=1}^{k} sweight_{1i} + \sum_{i=1}^{k} sweight_{2i}}{\sum_{i=1}^{|T_1|} weight_{1i} + \sum_{i=1}^{|T_2|} weight_{2i}} \quad (4)$$

The similarity between the two tables is always given as a normalized value between zero and one, as shown in equation (5),

$$0 \leq sim(T_1, T_2) \leq 1 \quad (5)$$

The similarity metric is used for modifying the KNN algorithm into the table based version as the approach to the text categorization.

Let us make some remarks on the similarity metric between tables which is described in this section. The function was defined for generating a set of words from the table. The shared table where each entry consists of a shared word and its dual weights is constructed from the two input tables. The similarity between tables is computed as the rate of the weight sum of the shared words to one of the all words in both tables. The similarity metric will be utilized for modifying the KNN algorithm into the table based version which is described in the next section.

C. Proposed Version of KNN

This section is concerned with the table based version of the KNN algorithm. In the previous section, we described the similarity between tables which is used for modifying the KNN algorithm. In the modified KNN algorithm, a novice text is encoded into a table, and its similarities with the sample tables are computed by the similarity metric. The proposed version of the KNN algorithm is adopted for implementing the text classification system which will be mentioned in the next section. This section is intended to describe the modified KNN algorithm as the approach to the text categorization.

Figure 5 illustrated that the similarities of a novice table with the sample tables are computed for selecting nearest neighbors. A novice text is encoded into the table, T_{nov}, the predefined categories are notated by $C = \{c_1, c_2, \ldots, c_{|C|}\}$, and the training set which consists of n sample tables which represent the sample texts is notated by $Tr = \{(T_1, y_1), (T_2, y_2), \ldots, (T_n, y_n)\}$, where T_i is a sample table, and $y_i \in C$. The similarities of the novice table, T_{nov} with the sample tables, $T_1, T_2, \ldots, T_n$, are computed by equation (4), as $sim(T_{nov}, T_1), sim(T_{nov}, T_2), \ldots, sim(T_{nov}, T_n)$ in the proposed KNN algorithm. The similarity between the novice table, T_{nov}, and a sample table, is given as a normalized value between zero and one, as shown in equation (5). The similarities, $sim(T_{nov}, T_1), sim(T_{nov}, T_2), \ldots, sim(T_{nov}, T_n)$ are ranked by their values for selecting nearest neighbors.

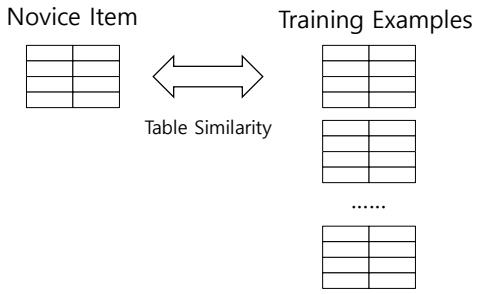

Figure 5. Similarities of a Novice Table with Sample Ones

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 (4) are ranked into ones, $sim(T_{nov}, T'_1), sim(T_{nov}, T'_2), \ldots, sim(T_{nov}, T'_n)$. The K items with their highest similarities with the novice item are selected as its nearest neighbors, as expressed in equation (6),

$$Near(K, T_{nov}) = \{T'_1, T'_2, \ldots, T'_K\} K \ll N \qquad (6)$$

As an alternative way, we may consider selecting items with their higher similarities than a given threshold. We use the nearest neighbors,$T'_1, T'_2, \ldots, T'_K$ from the training examples, for deciding the label of the novice table, T_{nov}.

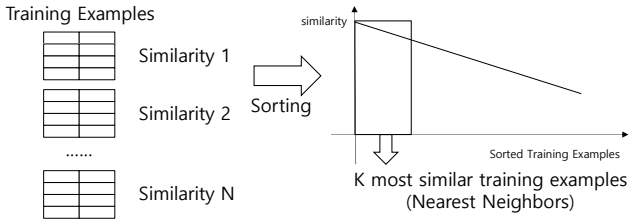

Figure 6. Selection of Nearest Neighbors from Training Examples

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, $Ne = \{T'_1, T'_2, \ldots, T'_K\}$, and the function for weighting a nearest neighbor by a category is defined as equation (7),

$$w(C_i, T'_j) = \begin{cases} 1 & \text{if } T'_j \in C_i \\ 0 & \text{otherwise} \end{cases} \qquad (7)$$

For each category, the number of nearest neighbors which belong it is counted as shown in equation (8),

$$Count(C_i, Ne) = \sum_{j=1}^{K} w(C_i, T'_j) \qquad (8)$$

The label of a novice item is decided by the label with the majority of the nearest neighbors, $C_{\max}$, as shown in equation (9),

$$C_{\max} = \underset{i=1}{\overset{|C|}{\operatorname{argmax}}} Count(C_i, Ne) \qquad (9)$$

The function, $w(C_i, T'_j)$ may be expanded into $w(C_i, T'_j, T_{nov})$ by augmenting the novice item, if the weight is dependent on the distance between the nearest neighbor and the novice item.

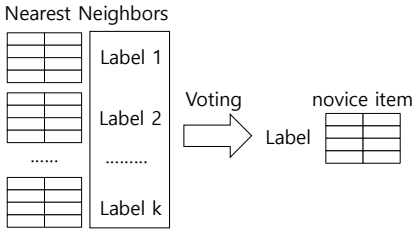

Figure 7. Voting Labels of Training Examples for deciding One of Novice Example

Let us some remarks on the proposed version of the KNN algorithm which is described in this section. Texts are encoded into tables for using the proposed KNN version for the text categorization. The similarities of a novice table with the sample tables is computed by the similarity metric which was described in Section III-B. The sample tables

are ranked by their similarities with the novice one, and the K sample tables are selected as its nearest neighbors. The labels of the nearest neighbors are voted for deciding the label of the novice table.

D. Text Categorization System

This section is concerned with the system architecture and the execution flow of the text categorization system. Texts are encoded into tables by the process which is described in Section III-A, and the KNN algorithm which is described in Section III-C is adopted. In this section, we present the system architecture and the execution flow for designing the text categorization system. In next research, we consider the implementation of the proposed system in Java or Python. This section is intended to describe the system architecture and the execution flow.

The collection of the sample texts for building the training set is illustrated in Figure 8. The topics are predefined as a list of topic 1, topic 2, ... , topic M; in implementing the system, it is assumed that the text classification belongs to the flat classification where the predefined categories are given as a list. For each topic, texts about it are collected, and encoded into tables by the process which was described in Section III-A. The M groups of tables are shown in the bottom of Figure 8, as the training set. The hierarchical text categorization will be considered in implementing the next version of the text categorization system.

The system architecture of the text categorization system which consists of the encoding module, the similarity computation module, and the voting module, is illustrated in Figure 9. The encoding module is for encoding texts into tables by the process which was described in Section III-A. The similarity computation module is as the core part of the system for computing the similarities of a table which represents a novice text and with the tables which represent the sample texts, and selecting ones with their highest similarities as the nearest neighbors. The voting module is for deciding the label of the novice item by voting ones of the nearest neighbors. In the proposed system, unlabeled texts are classified by the KNN algorithm which was described in Section III-C.

The execution flow of the text categorization system is illustrated in Figure 10. Both the sample texts and a novice text are encoded into tables. The similarities of the novice text with the sample texts are computed by the similarity metric which is described in Section III-B. Some with their highest similarities are selected as the nearest neighbors, and their labels are voted for deciding the label of the novice one. The category of the novice text is generated as the output in the execution flow which is presented in Figure 10.

Let us make some remarks on the system architecture and the execution flow of the text categorization system in Figure 9 and 10. We proposed encoding of texts into tables and the similarity metric between tables. The KNN algorithm

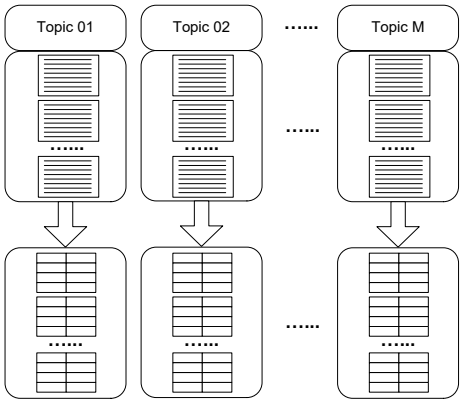

Figure 8. Collection of Sample Texts

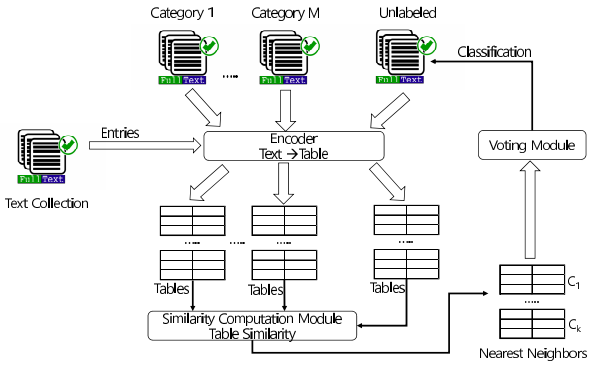

Figure 9. System Architecture

is modified as the approach to the text categorization by adopting the similarity metric for computing the similarity between a novice item and a sample one. The system architecture and the execution flow which are presented in this section indicate staying in the general design of the system. In the next research, we consider the detail design and the coding of the system.

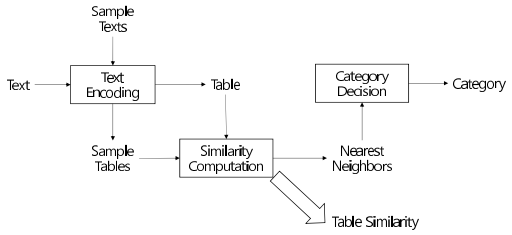

Figure 10. Execution Process

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 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 [16]. 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'.

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

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

of the four categories which are presented in Table I, and they are encoded into numerical vectors and tables. 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.

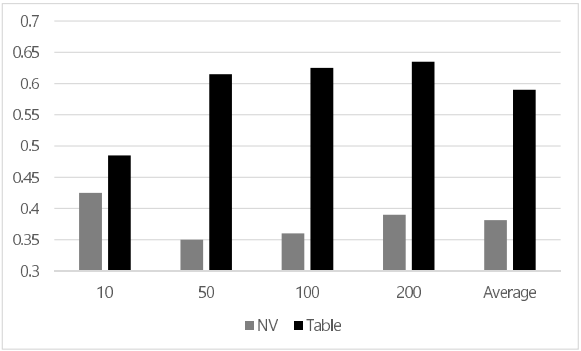

Figure 11. Results from Classifying Texts in Text Collection: NewsPage.com

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 0.64. The proposed version of KNN algorithm works strongly better in the all input sizes. The performance difference between the two versions is outstanding in the two input sizes, 50 and 100. From this set of experiments, 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 tables 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.

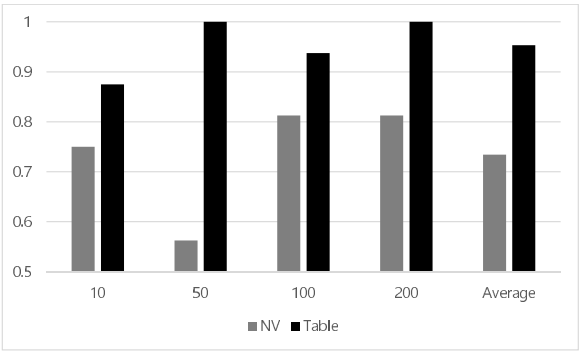

Figure 12. Results from Classifying Texts in Text Collection: Opiniopsis

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 works better than the traditional one in the all input sizes. It shows the perfect results in the input sizes: 50 and 200. From this set of experiments, we conclude that the proposed version works outstandingly 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.

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

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 tables 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 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.

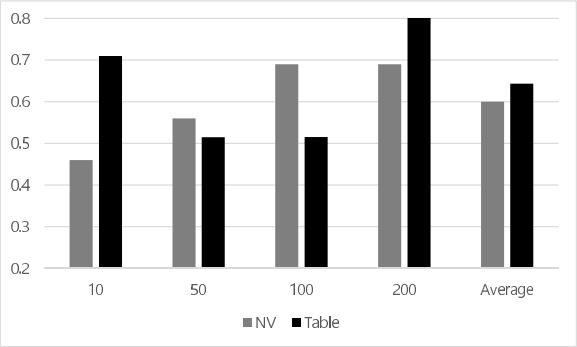

Figure 13. Results from Classifying Texts in Text Collection: 20NewsGroup I

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 13. The accuracies of the both versions range between 0.45 and 0.8. The proposed version shows its strongly better performance in the input size, 10. However, it is leaded by the traditional version in the others. From this set of experiments, we conclude that the proposed version keeps its better performance, in averaging the achievements of the four input sizes, in spite of its leaded performance.

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.

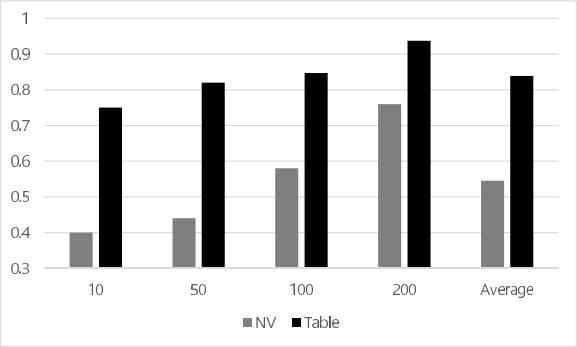

Figure 14. Results from Classifying Texts in Text Collection: 20NewsGroup II

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 0.92. The proposed version shows its better performance in all of the four input sizes. Even if the performance of both versions is proportional to the input size, the proposed version shows its better tolerance to the smaller input sizes. 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.

We need to consider the remaining tasks for doing the further research. The proposed approach needs to be validated and applied to the classifications of texts in the specific domains such as medicine, law, engineering, and so on, rather than in the various domains. We need the semantic relations among different words in the tables in computing the similarity between them. We may install the process of optimizing the weights of words in the tables as a meta-learning task. Adopting the proposed approach, we may implement the text categorization system as a real program or a module.

References

[1] T. Jo and N. Japkowicz, “Text Clustering using NTSO”, 558-563, The Proceedings of IJCNN, 2005.

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

[3] 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, 2007.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] T. Jo, “Table based Matching Algorithm for Soft Categorization of News Articles in Reuter 21578”, 875-882, Journal of Korea Multimedia Society, Vol 11, No 6, 2008.

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

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

[12] 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.

[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] 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.

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

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

[17] T. Jo, “Table based KNN for Categorizing Words”, 696-700, The Proceedings of 18th International Conference on Advanced Communication Technology, 2016.

[18] T. Jo, “Graph based KNN for Optimizing Index of News Articles”, 53-62, Journal of Multimedia Information System, Vol 3, No 3, 2016.

[19] 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.

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

[21] T. Jo, “Table based AHC for Text Clustering”, 133-138, The Proceedings of 13th International Conference on Data Mining, 2017.

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

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

[24] 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.

[25] T. Jo, “Table based K Nearest Neighbor for Word Categorization in News Articles”, 1214-1217, The Proceedings of 25th International Conference on Computational Science & Computational Intelligence, 2018.

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

[27] T. Jo, “Modification into Table based K Nearest Neighbor for News Article Classification”, 49-50, The Proceedings of 1st International Conference on Advanced Engineering and ICT-Convergence, 2018.

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

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

[30] 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.

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

[32] 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.

[33] T. Jo, ”Text Mining: Concepts, Implementations, and Big Data Challenge”, Springer, 2019.

[34] T. Jo, “Applying Table based AHC Algorithm to News Article Clustering”, 8-11, The Proceedings of International Conference on Green and Human Information Technology, Part I, 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, “Graph based Version for Clustering Texts in Current Affair Domain”, 171-174, The Proceedings of 15st International Conference on Data Science, 2019.

[37] T. Jo, “Using Table based Version of K Nearest Neighbor for Classify Words Semantically”, Unpublished, 2020.

[38] T. Jo, “Similarity Metric between Tables for Modifying AHC Algorithm in Text Clustering”, unpublished, 2020.

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