Using Table based Version of K Nearest Neighbor for Classifying Words Semantically

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 table as its input data and is applied to the word 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 words, modify the KNN algorithm by replacing the exiting similarity metric by the proposed one, and apply it to the word categorization. The proposed KNN is empirically validated as the better approach in categorizing words in news articles and opinions. In using the table based KNN algorithm, it is easier to trace results from categorizing words.**

***Keywords*-Word Categorization, Table Similarity, Table based KNN**

I. INTRODUCTION

Word categorization refers to the process of classifying words into one or some of the predefined categories. Its preliminary task is to predefine a list of categories and allocate sample words to each of them. The sample words are encoded into their structured forms and they are learned to build the classification ability. Novice words are encoded into the structured forms and classified into their own category or categories. Even if other types of word categorization are available, the scope of this research is restricted to only hard word categorization which allows to classifying each word into only one category.

Let us consider some problems in encoding words into numerical vectors as the challenges of this research. In encoding so, many features which are given text identifiers are required for the system robustness[29]. Because each numerical vector which represents a word has its sparse distribution, there is very little discrimination among feature vectors for computing their distances[25]. If we use grammatical features for encoding words into numerical vectors, it becomes difficult to implement the encoding process. Therefore, this research challenges the problems by encoding words into tables instead of numerical vectors.

Let us mention some ideas as what this research tries to propose as the solutions to the above problems. In this research, each word is encoded into tables each of which consists of entries of texts including the word and its weights in them. The similarity measure between two tables is defined as a normalized value between zero and one. The KNN (K Nearest Neighbor) is modified into the version where a table is given as its input data. The modified version is applied to the text categorization as its approach.

Let us consider some benefits which are expected from this research. The table which is proposed as the text representations may be expected to be more compact than numerical vectors, because they need only much less than 100 entries for maintaining the system robustness. We expect more discrimination among tables than among numerical vectors because the sparse distribution is not available in the tables. We obtain both better performance and more stability from the proposed KNN version, because the defined similarity measure is always given as a normalized value between zero and one. The table size is given as the external parameter and should be optimized between the reliability and the computation speed.

Let us mention the organization of this research. In Section II, we explore the previous works which are relevant to this research. 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 significances of this research and the remaining tasks as the conclusion.

II. PREVIOUS WORKS

This section is concerned with the previous works which are relevant to this research. In Section II-A, we present the modernized KNN versions to text mining tasks. In Section II-B, we survey the schemes of encoding texts or words into structured data. In Section II-C, we describe the previous machine learning algorithms which receive alternative structured data such as tables and string vectors to numerical vectors. Therefore, in this section, we provide the history about this research, by surveying the relevant previous works.

A. Word Categorization and its Derived Tasks

This section is concerned with the previous cases of applying the modified KNN algorithm to the word categorization and its derived tasks. This section focuses on the semantic word categorization which is distinguished from the POS (Part of Speech) tagging and where each word is categorized by its meaning. We present other types of KNN algorithms which use alternative structured types, as well as tables. We mention the keyword extraction and the index optimization as the special type of word categorization, and present the cases of applying the KNN algorithms to the tasks. This section is intended to explore the previous cases of applying the special types of KNN to the word categorization and its derived tasks.

Let us survey the modernized version of KNN algorithm which are applied to the word classification. The feature similarity was considered in computing the similarity between a novice item and a training example by Jo in 2018 [9]. A word was encoded into a string vector, instead of a numerical vector, in using the KNN algorithm for the topic based word classification [10]. The modernized KNN version where a word is encoded into a graph for improving the performance was proposed [11]. In the above literatures, the fact that the classification performance is improved by modernizing the KNN algorithm, was presented.

The keyword extraction is derived from the word categorization as the binary classification, and let us survey the cases of applying the modernized KNN algorithm to the task. The KNN algorithm which uses the similarity metric which considers the similarities among features were applied to the keyword extraction [12]. The KNN algorithm which classifies a string vector directly was applied to the keyword extraction [13]. The results from applying the KNN algorithm which classifies a graph directly were successful [14]. In these literatures, the keyword extraction was mapped into the task where each word from a text is classified into keyword or non-keyword.

One more task, index optimization, is derived from the word categorization, and let us explore the cases of applying the modernized KNN algorithm to the task. The KNN algorithm which considers the feature similarities was used for the index optimization [15]. The KNN algorithm which classifies a string vector directly was applied to the same task [16]. There was the case of applying the KNN algorithm which is modernized into the version which processes graphs directly [7]. The index optimization is mapped into a classification task where each word is classified into one of expansion, inclusion, and removal, depending on its importance.

Let us mention some points of this research which is distinguished from the literatures which were surveyed above. We surveyed the KNN version which was modernized with the different directions and applied to the word categorization and its derived tasks. We mentioned the three modernized versions of KNN algorithm: the version which considers the feature similarities in computing a similarity between a training example and a novice item, the version which receives directly a string vector, and the version which processes graphs directly. The modernized version of KNN algorithm which is proposed in this study is one which classifies a table directly. In this study, we apply the proposed version to the word categorization.

B. Word and Text Encoding

This section is concerned with surveying the schemes of encoding words and texts into structured forms which are alternatives to numerical vectors. Some issues were previously pointed out in encoding texts and words into numerical vectors. In previous works, it was proposed that texts or words should be encoded into tables, string vectors, or graphs, to solve the issues. In the previous section, it was mentioned that the KNN algorithms which were modernized so were applied to the word categorization and its related tasks. This section is intended to survey the cases of encoding texts or words into non-numerical vectors.

Let us mention the previous cases of encoding words or texts into tables. In using the AHC algorithm for clustering semantically words, they were encoded into tables [17]. Texts were also encoded into tables for modifying the KNN algorithm as the approach to the text categorization [18]. The AHC algorithm was modernized so for clustering texts [21]. Therefore, we mentioned the previous cases of encoding texts or words into tables in other tasks.

Let us consider the previous cases of encoding them into string vectors. Words were encoded into string vectors whose elements are text identifiers in using the AHC algorithm for the word clustering [19]. Texts were encoded into string vectors whose elements are words in using the KNN algorithm for the text categorization [20]. The AHC algorithm which clusters string vectors directly was proposed as the approach to the text clustering [22]. The above literatures present the previous cases of encoding words or texts into string vectors.

Encoding raw data into graphs was tried by influence of the social mining [2]. It was proposed that words should be encoded into graphs in using the AHC algorithm for the word clustering [8]. It was proposed that texts should be encoded into graphs in using the KNN algorithm for the text categorization [23]. In using the AHC algorithm for the text clustering, it was proposed that texts should be encoded so [24]. In the above literatures, we present the previous cases of mapping raw data into graphs.

W mention on the three schemes of encoding texts or words for using the machine learning algorithms. We adopt the first encoding scheme where words are encoded into tables for implementing the topic based word categorization system. Words are represented into tables from the inverted index where each word is linked to a list of texts including itself. We define the similarity metric between tables for modernizing the KNN algorithm as the approach to the word categorization. The task to which we apply the proposed approach is the word categorization, the KNN algorithm is modified into the version which processes tables directly, and this research will be distinguished from the above literatures.

C. Non-Numerical Vector based Machine Learning Algorithms

This section is concerned with the previous works on the machine learning algorithms which process non-numerical vectors directly. In the previous section, we surveyed the previous works on encoding words or texts into alternative structures in using the KNN algorithm and the AHC algorithm. Now, we mention the three machine learning algorithms, the string kernel based Support Vector Machine, table matching algorithm, and Neural Text Categorizer, which processes non-numerical vectors, as the approach to the text categorization. Among them, the third was used for processing Arabian text processing and mentioned as one of innovative classification, previously. This section is intended to explore the previous works which are involved in the three machine learning algorithms which were applied to the text categorization.

The string kernel was proposed as a kernel function in using the Support Vector Machine for the classification task. It was initially mentioned as the solution to the problems in encoding texts into numerical vectors by Lodhi et al in 2002 [28]. Subsequently, it was applied to the protein classification where proteins are given as strings by Leslie et al in 2004 [27]. The string kernel used for processing semantically sentences instead of entire full texts by Kate and Mooney in 2006 [26]. The Support Vector Machine with the string kernel was not successful in the text classification, but successful in the protein classification.

The table matching algorithm was proposed as a classification algorithm where raw data is encoded into tables, in the previous works. In 2007, by Jo and Cho, the table based matching algorithm was initially proposed for implementing a text classification system [25]. It was applied to the soft categorization of texts where it is allowed to assign more than one topic to each text, in 2008 [3]. The table matching algorithm was improved into a more stable version in implementing the text classification system, in 2015 [6]. Texts were encoded into tables in the table matching algorithm which was proposed in the above literatures as the approach to the text classification.

The neural network model which was specialized for the text categorization task and called Neural Text Categorizer, was invented. It was initially proposed by Jo in 2008 as the approach to the text categorization [4]. It was applied to both the hard categorization and the soft categorization in 2010 [5]. It was applied to classification of texts in Arabic by Abainia et al. in 2015 [1]. It was mentioned as an innovative neural network model by Vega and Mendez-Vazquez in 2016 [30].

We mentioned the three classifications, the string kernel based SVM, the table matching algorithm, and the Neural Text Categorizer which process non-numerical vectors directly. The trials which is presented from the three classification algorithms are intended for solving the issues in encoding raw data into numerical vectors by encoding them into other structured forms. In the proposed algorithm as the approach to the word categorization, words are encoded into tables. Texts which include the word indicate the entries of table for representing the word, and the weights in the table indicate relationship between a text and the word. In this research, the KNN algorithm will be modified into the table based version as the approach to the word categorization.

III. Proposed Approach

This section is concerned with the table based KNN (K Nearest Neighbor) which is the approach to the word categorization, and consists of the three sections. In Section III-A, we describe the process of encoding words into tables. In Section III-B, we cover the scheme of computing a similarity between two tables. In Section III-C, we mention the proposed version of KNN as the word categorization tool, and in Section III-D, present the architecture of the system which we try to implement by adopting the proposed KNN. Therefore, this section is intended to describe in detail the encoding scheme, the similarity computing method, and the proposed KNN for implementing the word categorization system.

A. Word Encoding

This section is concerned with the process of encoding words into tables. We surveyed the previous cases of encoding texts and words so in Section II-B and II-C. Each word is encoded into a table with the three steps which is shown in Figure 1, 2, and 3: corpus indexing, invert indexing, and term weighting. Each entry is given as a text identifier including the word and its weight in the table which represents it. This section is intended to describe the process of mapping a word into a table.

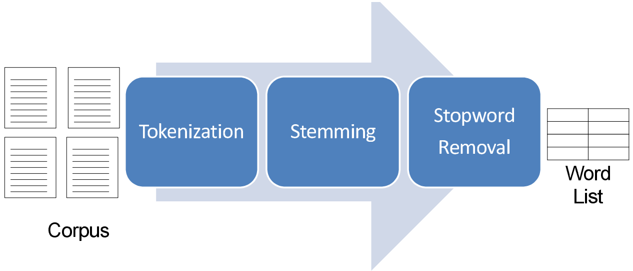

Figure 1. Overall Process of Word Indexing

The process of indexing a corpus into a list of words is illustrated in Figure 1. Each sentence in the corpus is segmented into tokens by white spaces, punctuation marks, and other special characters. Each token is mapped into its own grammatically root form in the stemming. Stop words such as conjunctions, prepositions and articles which function only grammatically and irrelevantly to the contents are removed. A list of words is generated as the output from this process.

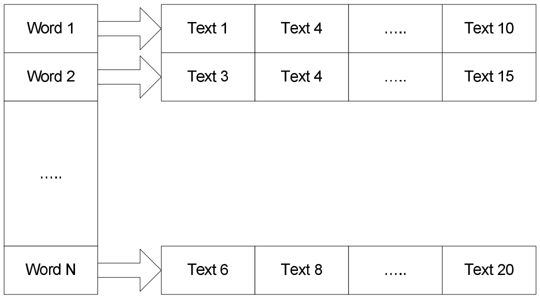

Figure 2. Inverted Index

The inverted index which is constructed after indexing the corpus is illustrated in Figure 2. Each text is expected to be linked with its own words after indexing the corpus. However, each word is linked with a list of texts which includes itself. The index structure where each text is linked with its own words is converted into one where each word is linked with its relevant texts. The fact that the axis is given to words rather than texts is the reason of calling what is presented in Figure 2, inverted index.

The process of assigning a weight to each text identifier in the table illustrated in Figure 3. A list of text identifiers which are related with the word is retrieved by the inverted indexing. We adopt the TF-IDF (Term Frequency and Inverse Document Frequency) weight as the relationship between a text and the word, and the weight is computed for each entry by the equation which is presented in Figure 3. The TF-IDF weight is proportional to the frequency in the text, but inversely proportional to the document frequency in the corpus. The preparation of the corpus is required for computing the TF-IDF weights.

In this research, a word is encoded into the three steps which are presented in Figure 1, 2, and 3. A table where constant weights are assigned to all entries is viewed as an unordered set of text identifiers. Each weight which is assigned to a text identifier indicate relationships between the word and a text identifier. A table as the unordered set of entries each of which consists of a text identifier and its weight represent a word. The entry is expanded into one with a text identifier and its multiple weights in using multiple schemes of weighting text identifiers.

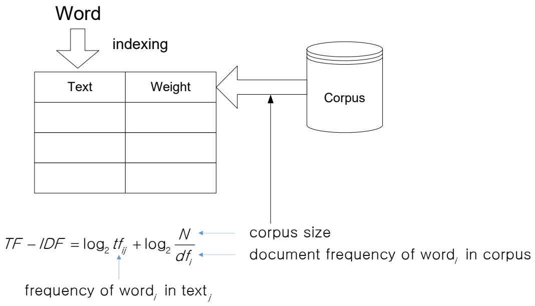
Word
indexing
Text
Weight
Corpus
$TF-IDF = \log_2 tf_{ij} + \log_2 \frac{N}{df_i}$
corpus size
document frequency of word$_i$ in corpus
frequency of word$_i$ in text$_j$

Figure 3. Text Weighting

B. Table Similarity

This section is concerned with the computation of similarity metric between two tables. In the previous section, we mentioned the process of mapping words into tables. In this section, we define the similarity metric between two tables for modifying the KNN algorithm into the version which processes tables directly. A table is expressed as a set of entries each of which consists of a text identifier and its weight. This section is intended to describe the definition and the computation of the similarity metric between two tables.

Let us mention the function of a table for mapping a table into an item set. A table is expressed as a set of entries as shown in equation (1),

$$T = \{(text\_id_1, weight_1), (text\_id_2, weight_2), \ldots, (text\_id_{|T|}, weight_{|T|})\} \tag{1}$$

where $text\_id_i$ is a text identifier which include the word and $weight_i$ is its weight in the text identified by $text\_id_i$. The table function is defined for generating a list of text identifiers as expressed in equation (2),

$$F(T) = \{text\_id_1, text\_id_2, \ldots, text\_id_{|T|}\} \tag{2}$$

The elements in the set, $F(T)$, is given text identifiers which include the word which is represented by the table, T. The function will be used for computing the similarity between two tables.

Let us mention the process of computing the similarity between two tables which represent words. The two tables are expressed as two sets of entries in equation (3) and (4),

$$\begin{aligned} T_1 = \{(text\_id_{11}, weight_{11}), (text\_id_{12}, weight_{12}), \\ \ldots, (text\_id_{1|T|}, weight_{1|T|})\} \end{aligned} \tag{3}$$

$$\begin{aligned} T_2 = \{(text\_id_{21}, weight_{21}), (text\_id_{22}, weight_{22}), \\ \ldots, (text\_id_{2|T|}, weight_{2|T|})\} \end{aligned} \tag{4}$$

The two tables are mapped into the sets of text identifiers which are shown in equation (5) and (6), by applying the table function to equation (3) and (4),

$$F(T_1) = \{text\_id_{11}, text\_id_{12}, \ldots, text\_id_{1|T|}\} \tag{5}$$

$$F(T_2) = \{text\_id_{21}, text\_id_{22}, \ldots, text\_id_{2|T|}\} \tag{6}$$

The set of shared text identifiers which is shown in equation (7) is obtained by applying the intersection to equation (5) and (6),

$$F(T_1) \cap F(T_2) = \{stext\_id_1, stext\_id_2, \ldots, stext\_id_k\} \tag{7}$$

The shared table is constructed by taking their weights from the two tables, T_1 and T_2 as shown in equation (8),

$$\begin{aligned} ST = \{(stext\_id_1, weight_{11}, weight_{21}), \ldots, \\ (stext\_id_k, weight_{1k}, weight_{2k})\} \end{aligned} \tag{8}$$

In equation (8), $weight_{1i}$ indicates the weight from the table, T_1, and $weight_{2i}$ indicates the weight from the table, T_2 to the text identifier, $stext\_id_1$.

Let us mention the process of computing the similarity between two tables after extracting the shared entries. The weights of the two tables are given as sums of entry weights, as expressed in equation (9) and (10),

$$W(T_1) = \sum_{i=1}^{|T_1|} weight_{1i} \tag{9}$$

$$W(T_2) = \sum_{i=1}^{|T_2|} weight_{2i} \tag{10}$$

The dual weight sums in the shared table, ST, are defined as equation (11) and (12),

$$W_1(ST) = \sum_{i=1}^{k} sweight_{1i} \tag{11}$$

$$W_2(ST) = \sum_{i=1}^{k} weight_{2i} \tag{12}$$

The similarity between the tables, T_1 and T_2 is computed by equation (13),

$$sim(T_1, T_2) = \frac{W_1(ST) + W_2(ST)}{W(T_1) + W(T_2)} \tag{13}$$

The similarity between tables is always given as normalized value between zero and one.

Above, we mentioned the similarity between two tables as a normalized value between zero and one. If the two tables are identical to each other as shown in equation (14),

$$T_1 = T_2 \tag{14}$$

the similarity between them is 1.0, as shown in equation (15),

$$\begin{aligned} sim(T_1, T_2) &= \frac{W_1(ST) + W_2(ST)}{W(T_1) + W(T_2)} \\ &= \frac{2W_1(ST))}{2W(T_1)} = \frac{2W_1(T_1))}{2W(T_1)} = 1.0 \end{aligned} \tag{15}$$

If the two tables are completely different from each other as shown in equation (16),

$$F(T_1) \cap F(T_2) = \oslash, |ST| = 0 \tag{16}$$

the similarity between them is zero, as shown in equation (17),

$$\begin{aligned} sim(T_1, T_2) &= \frac{W_1(ST) + W_2(ST)}{W(T_1) + W(T_2)} \\ &= \frac{0}{W(T_1) + W(T_2)} = 0.0 \end{aligned} \tag{17}$$

The similarity between two tables is given as a normalized value between zero and one, as shown in equation (18),

$$\begin{aligned} ST \subseteq T_1, ST \subseteq T_2 \\ W_1(ST) + W_2(ST) \leq W(T_1) + W(T_2) \end{aligned} \tag{18}$$

The similarity threshold is set between zero and one in modifying machine learning algorithms using the operation.

C. Proposed Version of KNN

The proposed version of the KNN algorithm is illustrated in Figure 4. Training example are given as tables, and the process of encoding words so was described in Section III-A. The similarity metric between tables which was described in Section III-B will be used for selecting nearest neighbors. A label of each novice item is decided by voting ones of the selected nearest neighbors. This section is intend to describe the proposed version of the KNN algorithm and its variants.

The essential step of classifying an item by the KNN algorithm is to select nearest neighbors as the references for deciding its label. As shown in Figure 4, it is assumed that the training examples and a single novice item are given as tables. Its similarities with the training examples are computed by the similarity metric which is described in Section III-B. The training examples are ranked by their

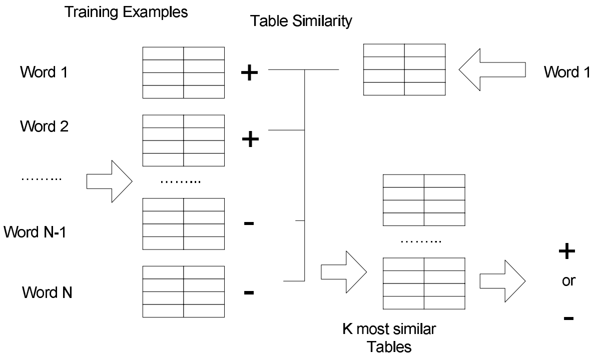

Figure 4. Proposed KNN Algorithm

similarities and K highest similar training examples are selected as its nearest neighbors. We adopt the rank based selection in selecting the nearest neighbors in this research.

Let us mention the process of voting the labels of the nearest neighbors for deciding one of a novice item. We notate the set of nearest neighbors of the novice item, T , whose elements are given as tables and their target labels, by equation (19),

$$\begin{aligned} Ne_k(T) = \{(T_1, y_1), (T_2, y_2), \ldots, (T_k, y_k)\}, \\ y_i \in \{c_1, c_2, \ldots, c_m\} \end{aligned} \tag{19}$$

where $c_1, c_2, \ldots, c_m$ are the predefined categories and k is the number of nearest neighbors. The number of the nearest neighbors which are labeled with the category,c_i is notated by $Count(Ne_k(T), c_i)$. The label of the novice item, T, is decided by the majority of categories in the nearest neighbors, as expressed by equation (20),

$$c_{\max} = \operatorname*{argmax}_{i=1}^{m} Count(Ne_k(T), c_i) \tag{20}$$

The external parameter,k, is usually set as an odd number for avoiding the possibility of largest number of nearest neighbors to more than one category.

Let us mention the weighted voting of labels of nearest neighbors as the alternative scheme to the above. Assuming that the similarity between two tables as a normalized value between zero and one, and we may use the similarities with the nearest neighbors, $sim(T, T_1), sim(T, T_2), \ldots, sim(T, T_k)$ as weights, $w_1, w_2, \ldots, w_k$ by equation (21),

$$w_i = sim(T, T_i) \tag{21}$$

indicates the similarity of a novice table with the ith nearest neighbor. The total weight of nearest neighbors which labeled with the category, c_i by equation (22),

$$Weight(Ne_k(T), c_i) = \sum_{T_j \in c_i}^{k} w_j \tag{22}$$

The label of the novice item, T, is decided by the category which corresponds to the maximum sum of weights as shown in equation (23),

$$c_{\max} = \operatorname*{argmax}_{i=1}^{m} Weight(Ne_k(T), c_i) \tag{23}$$

When the weights of nearest neighbors are set constantly, equation (23) is same to equation (20), as expressed in equation (24),

$$Weight(Ne_k(T), c_i) = Count(Ne_k(T), c_i) \tag{24}$$

We described the proposed version of the KNN algorithm in this section. In using the proposed KNN algorithm, raw data is encoded into tables, instead of numerical vectors. The similarities of a novice item with the training examples are computed by the similarity metric which is defined in Section III-B. The rank based selection is adopted as the scheme of selecting nearest neighbors among training examples. Because we are interested in the comparison of the traditional version and the proposed version as the ultimate goal, we use the unweighted voting in the experiments which are covered in Section IV.

D. *Word Classification System*

This section is concerned with the word classification system which adopts the table based KNN algorithm as the approach. We described the proposed version of KNN algorithm as the approach to the word classification in Section III-C. The preliminary tasks for doing the classification task is to predefine categories as a list of a hierarchical form and to collect sample labeled words. Words are encoded into tables and the KNN algorithm is applied to the word classification. This section is intended to describe the word classification system which is implemented in this study.

The sample words are illustrated for implementing the topic based word classification by the proposed KNN algorithm in Figure 5. The topics are predefined as topic 1, topic 2, . . ., topic M. The N words are allocated for each topic as the sample words. The balanced distribution over the categories is necessary for preventing the bias toward a particular topic. $M \times N$ sample words are encoded into tables by the process which is mentioned in Section III-A.

The entire architecture of the proposed word categorization system is illustrated in Figure 6. The sample words which are labeled with one of M categories and the unlabeled ones as novice items are encoded into tables. For each novice table, its similarities with the sample tables are computed by the metric which is mentioned in Section III-B, in the similarity computation module, and the k most similar sample tables are selected as the nearest neighbors. The label of the novice item is decided by voting ones of nearest neighbors in the voting module. This system consists of the three components: the encoding module, the similarity computation module, and the voting module.

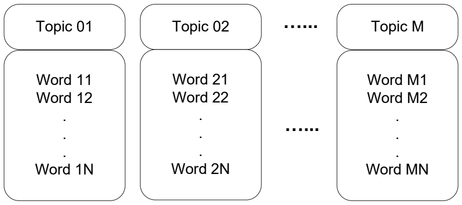

Figure 5. Sample Words

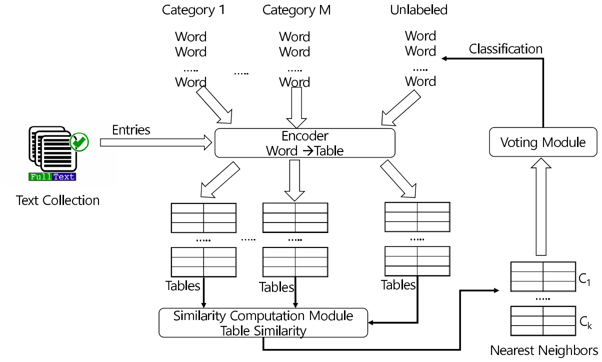

Figure 6. Proposed System Architecture

The execution process of the proposed system is illustrated in Figure 7. The sample words which are collected by the process mentioned above and the word which is given as the input are encoded into tables. Its nearest neighbors are extracted from the samples through the similarity computation module. The category of the novice word is decided by voting ones of the nearest neighbors. The category of the novice word is decided as the final output in the system.

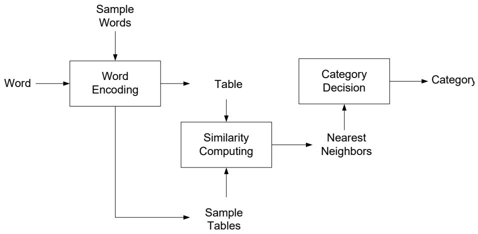

Figure 7. Execution Process of Proposed System

Let us make some remarks on the proposed system which is illustrated in Figure 6 as its architecture. Words are encoded into tables, instead of numerical vectors. Tables which represent novice words are classified directly by the proposed KNN algorithm. The classification performance is improved by what proposed in this research, as shown in Section IV. In the next research, we present the graphical user interface and the source code which are necessary for implementing the system as a complete one.

IV. Experiments

This section is concerned with the empirical experiments for validating the proposed version of KNN, and consists of the four sections. In Section IV-A, we present the results from applying the proposed version of KNN to the word categorization on the collection, NewsPage.com. In Section IV-B, we show the results from applying it for categorizing words from the collection, Opinosis. In Section IV-C, we mention the results from comparing the two versions of KNN with each other in categorizing words 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 from the collection, NewsPage.com, we gathered the words category by category as the labeled ones. Each word is allowed to be classified into only one of the four categories. In this set of experiments, we apply the traditional and proposed version of KNN to the classification task, without decompose it into binary classifications, and use the accuracy as the evaluation measure. In this section, we observe the performance of the both versions of KNN, by changing the input size.

In Table I, we specify NewsPage.com, which is the text collection as the source for extracting classified words in this set of experiments. The text collection was used in the previous works for evaluating approaches to text categorization [6]. In each category, we extract 375 important words for building the collection of labeled words for evaluating the approaches to word categorization. In each category, the set of 375 classified words is partitioned into the 300 words as training examples and the 75 words as test examples, as shown in Table I. We select words by their frequencies concentrated in the given category combined with subjectivity in building the word collection.

Table I
THE NUMBER OF TEXTS AND WORDS IN NEWSPAGE.COM

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

Let us mention the empirical process for validating the proposed approach to the task of word categorization. We extract the important words from each category in the above text collection, and encode them into numerical vectors. For each text example, the KNN compute its similarities with the 1200 training examples by the cosine similarity, and select the three most similar 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. The classification accuracy is computed by dividing the number of correctly classified test examples by total number of test examples, for evaluating the both versions of KNN.

Figure 8 illustrates the experimental results from categorizing the words using the both versions of KNN algorithm. The y-axis indicates the accuracy which is the rate of the correctly classified examples in the test set. Each group in the x-axis is the input size as the dimension of numerical vectors and the number of entries of tables. In each group, the gray and black bar indicate the performance of the traditional and proposed version of KNN algorithm, respectively. The most right group indicates the average over accuracies of the left four cases.

Let us make discussions on the results from doing the word categorization, using the both versions of KNN algorithm, as shown in Figure 8. The accuracy which are the performance measure of this classification task is in range between 0.24 and 0.32. The proposed version of KNN algorithm works better in the input sizes: 10 and 50. The proposed version matches with the traditional one in the input size, 100. In this set of experiments, we conclude that the proposed version works outstandingly better than the traditional one, in averaging over the four cases.

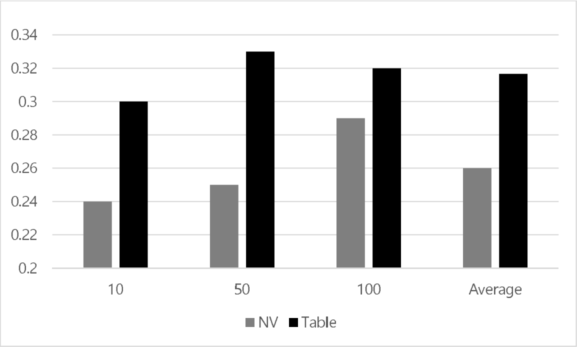

Figure 8. Results from Classifying Words in Text Collection: NewsPage.com

B. *Opinopsis*

This section is concerned with the set of experiments for validating the better performance of the proposed version on the collection: Opinosis. In this set of experiments, the three categories are predefined in the collection, and we gather words category by category as the classified ones. Each word is classified exclusively into one of the three categories. The given classification is not decomposed into binary classifications and the accuracy is used as the evaluation measure. In this section, we observe the performances of the both versions of KNN algorithm with the different input sizes in the collection, Opniopsis.

In Table II, we illustrate the text collection, Opinosis, which is used as the source for extracting the classified words, in this set of experiments. The collection was used in previous works, for evaluating the approaches to text categorization. We extract the 375 important words from each category as the collection of the classified words for evaluating the approaches to word categorization. In each category, as shown in Table 2, we partition the set of words into the 300 words as the training set and the 75 words as the test set. We select the words from the collection, depending on their frequencies which are concentrated on their own categories.

Table II
THE NUMBER OF TEXTS AND WORDS IN OPINIOPSIS

Category	#Texts	#Training Words	#Test Words
Car	23	300	75
Electronic	16	300	75
Hotel	12	300	75
Total	51	900	225

We perform this set of experiments by the process which is described in section IV-A. We extract the 300 important words by scanning individual texts in each category, and encode them into numerical vectors and tables, with the input sizes: 10, 50, and 100. For each test example, the both versions of KNN computes its similarities with the 900 training examples and select the three most similar training examples as its nearest neighbors. Each of the 225 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 9, we illustrate the experimental results from categorizing the words using the both versions of KNN on this collection. Like Figure 8, the y-axis indicates the accuracy and the x-axis does the group of two versions by an input size. In each group, the grey bar and the black bar indicate the results of the traditional version and the proposed version of KNN algorithm, respectively. In Figure 2, the most right group indicates the average over results of the left three groups. Therefore, Figure 9 presents the results from classifying the words into one of the three categories by both versions of KNN algorithm, on the collection, Opinosis.

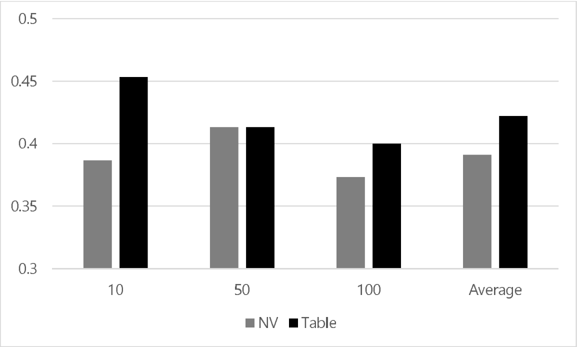

Figure 9. Results from Classifying Words in Text Collection: Opiniopsis

We discuss the results from doing the word categorization using the both versions of KNN algorithm, on Opinosis, shown in Figure 9. The accuracies of the both versions range between 0.35 and 0.45 in this task. The proposed version works better than the traditional one in the two input sizes: 10 and 100. It is comparable with the traditional version in the other: 50. From this set of experiments, we conclude that the proposed one works slightly better in averaging over the four cases.

C. 20NewsGroups I: General Version

This section is concerned with one more set of experiments where the better performance of the proposed version is validated empirically on the text collection: 20NewsGroups I. In this set of experiments, we predefine the four general categories, and gather words from the collection category by category as the classified ones. Each word is classified exclusively into one of the four categories. We apply the KNN algorithms directly to the given task without decomposing it into binary classification, and use the accuracy as the evaluation measure. Therefore, in this section, we observe the performance of the both versions of KNN algorithm, 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 1000 texts at random, and extract 375 important words from them as the labeled words. The 375 words are partitioned into the 300 words as the training examples and the 75 words as the test ones, as shown in Table III. In the process of gathering the classified words, they are selected by their frequencies which are concentrated in their corresponding categories.

Table III
The Number of Texts and Words in 20NewsGroups I

Category	#Texts	#Training Words	#Test Words
Comp	1000	300	75
Rec	1000	300	75
Sci	1000	300	75
Talk	1000	300	75
Total	4000	1200	300

The experimental process is identical is that in the previous sets of experiments. In each category, we extract the 375 important words 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 10, we illustrate the experimental results from categorizing words using the both versions on the broad version of 20NewsGroups. Figure 10 has the identical frame of presenting the results to those of Figure 1 and 2. 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. The performance is expressed as the accuracy of classifying words into one of the four categories. In this set of experiments, the classification task is not decomposed into binary classifications.

Let us discuss the results from doing the word categorization using the both versions on 20NewsGroups as shown in Figure 10. The accuracies of the both versions range between 0.28 and 0.49. The proposed version of KNN algorithm

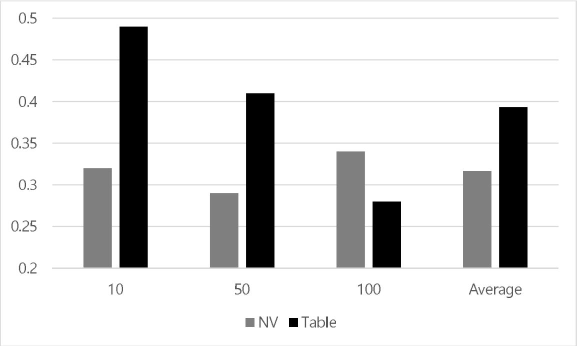

Figure 10. Results from Classifying Words in Text Collection: 20NewsGroup I

shows its better performances in the three of the four cases, but slightly less performance in the other. The inconsistent entries and the noisy values are the causes of degrading the performance of the proposed version, in the input size, 200. From this set of experiments, we conclude that the proposed version wins over the traditional one, in averaging over their four achievements, in spite of that.

V. Conlusion

Let us discuss the entire results from classifying words using the two versions of KNN algorithm. We compare the two versions with each other in the three collections. The proposed versions show its better results in all of the three collections. On the three collections, the accuracies of the traditional version range between 0.24 and 0.45, while, those of the proposed version range between 0.28 and 0.49. Finally, through the three sets of experiments, we conclude that the proposed version of KNN algorithm improves the word categorization performance, as the contribution of this research.

Let us consider the remaining tasks for doing the further research. We need to validate and customize the proposed research in the word categorization in one of specific domains: engineering, science, and medicine. Because various schemes of weighting words are available, more than one weight may be assigned to each word, so it need to be considered in computing the similarity between tables. Other machine learning algorithms may be modified as well as KNN into their table based versions. By adopting the proposed approach, we implement the word categorization system as a module of other programs or an independent program.

References

[1] 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.

[2] D.J. Cook and L.B. Holder, Mining Graph Data, Wiley, 2007.

[3] 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.

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

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

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

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

[8] T. Jo, “Encoding Words into Graphs for Clustering Word by AHC Algorithm”, 90-95, The Proceedings of 12th International Conference on Multimedia Information Technology and Applications, 2016.

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

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

[11] 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.

[12] T. Jo, “Extracting Keywords from News Articles using Feature Similarity based K Nearest Neighbor”, 68-71, The Proceedings of International Conference on Information and Knowledge Engineering, 2018.

[13] T. Jo, “Modifying K Nearest Neighbor into String Vector based Version for Extracting Keywords from News Articles”, 43-46, The Proceedings of International Conference on Applied Cognitive Computing, 2018.

[14] T. Jo, “Graph based K Nearest Neighbors for Keyword Extraction in Current Affair Domain”, 47-48, The Proceedings of 1st International Conference on Advanced Engineering and ICT-Convergence, 2018.

[15] T. Jo, “Index Optimization in News Articles using Feature Similarity based K Nearest Neighbor”, 106-109, The Proceedings of 17th Int’l Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government, 2018.

[16] T. Jo, “String Vector based Version of K Nearest Neighbor for Index Optimization in Current Affairs”, 47-50, The Proceedings of International Conference on Applied Cognitive Computing, 2018.

[17] T. Jo, “Using Table based AHC Algorithm for clustering Words in Domain on Current Affairs”, 1222-1225, The Proceedings of 25th International Conference on Computational Science & Computational Intelligence, 2018.

[18] 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.

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

[20] 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.

[21] 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.

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

[23] 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.

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

[25] 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.

[26] R. J. Kate and R. J. Mooney, “Using String Kernels for Learning Semantic Parsers”, 913-920, Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, 2006.

[27] C. S. Leslie, E. Eskin, A. Cohen, J. Weston, and W. S. Noble, “Mismatch String Kernels for Discriminative Protein Classification”, 467-476, Bioinformatics, Vol 20, No 4, 2004.

[28] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, “Text Classification with String Kernels”, 419-444, Journal of Machine Learning Research, Vol 2, No 2, 2002.

[29] F. Sebastiani, “Machine Learning in Automated Text Categorization”, 1-47, ACM Computing Survey, Vol 34, No 1, 2002.

[30] 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.