Text Segmentation based on Contents using String Vector based Version of K Nearest Neighbor

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 string vector as its input data and is applied to the text segmentation. The results from applying the string vector based algorithms to the text categorizations were successful in previous works, and the text segmentation is able to be viewed into a binary classification where each adjacent paragraph pair is classified into boundary or continuance. In the proposed system, a list of adjacent paragraph pairs is generated by sliding a text with the two sized window, each pair is classified by the proposed KNN version, and the boundary is put between the pairs which are classified into boundary. The proposed KNN version is empirically validated as the better approach in deciding whether each pair should be separated from each other or not in news articles and opinions. We need to define and characterize mathematically more operations on string vectors for modifying more advanced machine learning algorithms.**

I. Introduction

Text segmentation refers to the process of putting boundary in the location where the transition from a topic to another happens. Each text is partitioned into sentences or paragraphs and pairs of adjacent sentences or paragraphs are encoded into the structured forms. We prepare the set of sentence or paragraph pairs labeled with boundary or continuance as the samples and build the classification capacity by learning them. From a novice text, we generate pairs of adjacent sentences or paragraphs and put the boundary between them corresponding to pairs classified into boundary. In this research, we assume that the text segmentation is viewed as a binary classification and apply a supervised learning algorithm as the approach to the task.

Let us mention what provides the motivations for doing this research. After transforming speech texts into written texts, they need to be partitioned into sentences or paragraphs by the text segmentation process. The task may be transformed into a binary classification which is an instance of text categorization [**?**]. In the previous works, encoding texts into string vectors contributes to the improvement of text categorization performance [**?**]. Therefore, assuming that the text segmentation is an instance of text classification, we try to propose the sophisticated approach to text segmentation.

We mention some agenda which are proposed in this research as its ideas. We encode adjacent paragraph pairs or sentence pairs into string vectors and define the semantic similarity measure between two string vectors which corresponds to the cosine similarity between numerical vectors. Using the similarity measure, we modify the KNN (K Nearest Neighbor) into the string vector based version where a string vector is directly given as the input data. The modified version is used as the approach to the classification task which is mapped from the text segmentation. In this research, the text segmentation is regarded as the binary classification task to which the supervised learning algorithms are applicable.

Let us mention some advantages which are provided by this research. In this research, we are able to avoid completely the above problems in encoding texts into numerical vectors, by doing them into alternative structured forms. It is more efficient to encode texts into string vectors than into numerical vectors; in encoding texts into string vectors, only tens of elements is required for maintaining enough system robustness; while in doing them into numerical vectors, hundreds of elements with using feature selection schemes is required. String vectors which represent texts provide more transparency than numerical vectors; it is easier to view the contents referring to only representations which are string vectors. However, in order to modify more advanced machine learning algorithms, we need to define more advanced operations on string vectors.

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, 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. Applications to Text Mining Tasks

This section is concerned with the previous cases of applying the modernized KNN algorithm to the text segmentation and its similar tasks. In addition, the text categorization and the text summarization will be mentioned as ones similar as the text segmentation. The KNN algorithm which is the approach to the three tasks was modernized for solving the problems in encoding texts into numerical vectors. The successful results from applying the modernized version to the tasks were presented in the previous works. This section is intended to present the previous works with the successful results in applying the modernized version.

Let us explore the cases of using the modernized version of KNN algorithm for categorizing texts. The KNN algorithm was modified into the modernized version which solves the poor discrimination among the sparse vectors by considering the similarities among features, in using it for the text categorization [16]. Another modernized version of KNN algorithm which classifies a table directly, instead of a numerical vector, was proposed as the approach to the text categorization [8]. The KNN version which receives a graph as the input data was adopted for implementing the text categorization system [17]. The task which is mentioned in the above works is the source from which the text segmentation is derived together with the text summarization.

The text summarization is derived from the text categorization, as the similar task with the text segmentation. The modernized version of KNN algorithm which uses the similarity metric between numerical vectors, considering the feature similarities for implementing a text summarization system [9]. Another modernized version of KNN algorithm which processes tables directly, was applied to the text summarization [18]. The KNN version which classifies a graph directly was proposed as the approach to the text summarization [19]. In the above literatures, the text summarization was viewed as the binary classification where each paragraph is classified into summary or non-summary.

Let us explore the previous cases of applying the modernized versions of KNN algorithms to the task which is covered in this study. The KNN version which is modernized by considering the feature similarities in computing the similarity between vectors, was applied to the text segmentation [20]. Another modernized version which receives a table as its input data was used for implementing a text segmentation system [10]. The third modernized version which processes graphs directly, was proposed as the approach to the text segmentation [21]. The text segmentation was interpreted into the classification of an adjacent paragraph pair into boundary or continuance, in the above literatures, together with this study.

Let us mention some points which distinguish this research from ones which are surveyed above. We presented the previous cases of applying the three kinds of modernized KNN algorithms to the text segmentation and its related tasks. We mentioned the two related tasks: the text categorization from which the text segmentation is derived and the text summarization which is derived from the former. The proposed version of KNN algorithm is modernized in the different direction and deals with graphs which represent paragraph pairs. The text segmentation is mapped into the binary classification of adjacent paragraph pairs and the proposed version will be applied to the task, in this research.

B. Word and Text Encoding

This section is concerned with the previous cases of encoding words or texts into other structured forms which replace the numerical vectors. We have realized continually the issues in encoding them into numerical vectors for using the traditional learning algorithms. As solution to the issues, it has been proposed that they are encoded into other structured forms including string vectors. The tables and the graphs as well as string vectors will be mentioned as the replacement of numerical vectors for representing texts and words. This section is intended to survey previous cases of encoding them into the replacements.

Let us present the previous works on encoding words or texts into tables. Words were encoded into tables in using the KNN algorithm for the word categorization [13]. They were encoded so in using the AHC algorithm for the word clustering [14]. Texts were encoded into tables in using the AHC algorithm for the text clustering [22]. The previous works which are mentioned above become the cases of encoding raw data into tables.

Let us mention the previous works on encoding words or texts into string vectors. Words were encoded into string vectors in using the KNN algorithm for the topic based word classification [11]. In using the AHC algorithm for the semantic word clustering, words were encoded into string vectors [12]. Texts were encoded into string vectors in using the AHC algorithm for text clustering [23]. In the above literatures, we present the previous cases of encoding raw data into string vectors.

Let us survey the previous studies on encoding texts or words into graphs. Words were encoded into graphs for modernizing the KNN algorithm as the approach to the word categorization [7]. Words were encoded so for doing the AHC algorithm as the approach to the semantic word clustering [15]. Texts were encoded into graphs for doing the AHC algorithm as the approach to the text clustering [24]. In the literatures, we present the previous cases of encoding raw data into graphs.

We mentioned the three kinds of structured data as representations of words or texts in the previous works. We adopted the second kind of structured data, called string vectors, as representations of adjacent paragraph pairs, in this study. We define the similarity metric between string vectors, and modify the KNN algorithm into the version which classifies string vectors directly. We use the modified version for implementing the text segmentation system. We validate the performance of the modified KNN algorithm in the binary classification which is mapped from the text segmentation, comparing it with the traditional version.

C. Non-Numerical Vector based Machine Learning Algorithms

This section is concerned with the previous works on the supervised learning algorithms which process non-numerical vectors directly. In the previous section, we presented the cases of transforming texts or words into tables, graphs, string vectors, as non-numerical vectors. In this section, as the approaches to the text categorization, we mentioned the three supervised learning algorithms: string kernel based Support Vector Machine, Table based Matching Algorithm, and Neural Text Categorizer. In using the first, raw texts are used by themselves, in using the seconds they are encoded into tables, and in using the last they are encoded into string vectors. This section is intended to survey the previous works on the three approaches which are mentioned above.

Let us consider the string kernel as the way of avoiding problems in encoding texts into numerical vectors. The string kernel was initially proposed as the solution to the problems by Lodhi et al. in 2002 [28]. It was utilized for modifying the k means algorithm as the approach to the text clustering by Karatzonglou and Feinerer in 2006 [27]. The string kernel based SVM (Support Vector Machine) was applied to the sentence classification by Kate and Mooney in 2006 [26]. The string kernel which is mentioned in the above literatures is the kernel function of two raw texts based on characters in them.

Let us explore the previous works on the table based matching algorithm as another non-numerical vector based classification algorithm. It was initially proposed as the approach to the text categorization by Jo and Cho in 2008 [25]. It was applied to the soft text categorization where each text is allowed to be classified into more than one category in 2008 [2]. It was improved into the more robust and stable approach to the text categorization by Jo in 2015 [5]. Texts should be encoded into tables in using it for the text categorization tasks.

Let us consider the Neural Text Categorizer as the Perceptron like neural network model which is specialized for the text categorization. It was initially invented as the approach to the text categorization by Jo in 2008 [3]. Its better performance than those of the Na?ve Bayes and the SVM which are used as popular approaches to the task, was empirically validated in both the soft text categorization and hard one by Jo in 2010 [4]. It was applied to the classification of Arabian texts by Abainia et al. in 2015 [1]. It was mentioned as an innovative neural network model by Vega and Mendez-Vasquez in 2016 [29].

We surveyed the previous works on the non-numerical vector based approaches to the text categorization. Texts are encoded into tables or string vectors as replacements of numerical vectors. In this research, adjacent paragraph pairs which are given as texts are encoded into string vectors. The KNN algorithm is modified into the version which classifies string vectors directly as the approach to the text segmentation. The task is mapped into the classification of adjacent paragraph pair into continuance or boundary.

III. Proposed Approach

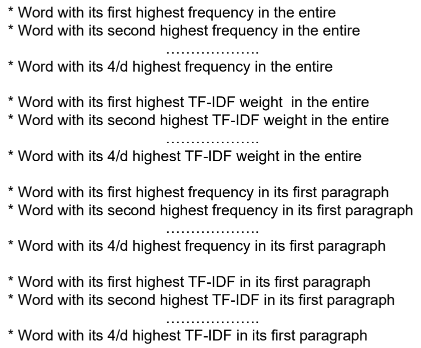
This section is concerned with encoding words into string vectors, modifying the KNN (K Nearest Neighbor) into the string vector based version and applying it to the text segmentation, and consists of the three sections. In section III-A, we deal with the process of encoding texts into string vectors. In section III-B, we describe formally the similarity matrix and the semantic operation on string vectors. In section III-C, we do the string vector based KNN version as the approach to the text segmentation. In Section III-D, we explain the architecture of the text segmentation system where the proposed KNN is adopted.

A. Text Encoding

This section is concerned with the transformation of texts into string vectors. In Section II-B and II-C, we explored the previous cases of encoding raw data into string vectors. In this study, a text is encoded into a string vector with the three steps: the feature definition, the feature matching analysis, and the word assignment. A string vector which represents a text consists of words in an order. This section is intended to describe the three steps which are presented in Figure 1-3.

The features which are defined for encoding a text into a string vector are illustrated in Figure 1. In defining them, it is assumed that in each text, its first paragraph is the key part and the dimension of the string vector is d. The group of d features is divided into the four subgroup by combining the text scope, entire text or first paragraph, with the relationship between a text or a word, frequency or TF-IDF (Term Frequency and Inverse Document Frequency). Words are ranked by their frequencies or their weights within each subgroup, until $d/4$. Features are manually defined in the current research, and it is necessary to automate it in the next research.

The process of analyzing the feature matching is illustrated as a pseudo code in Figure 2. A list of words which is resulted from indexing a text and one among the features which is shown in Figure 1 are given as the arguments of this



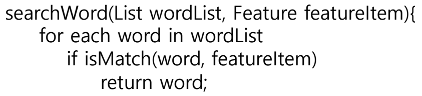
* Word with its first highest frequency in the entire
* Word with its second highest frequency in the entire
....
* Word with its 4/d highest frequency in the entire

* Word with its first highest TF-IDF weight in the entire
* Word with its second highest TF-IDF weight in the entire
....
* Word with its 4/d highest TF-IDF weight in the entire

* Word with its first highest frequency in its first paragraph
* Word with its second highest frequency in its first paragraph
....
* Word with its 4/d highest frequency in its first paragraph

* Word with its first highest TF-IDF in its first paragraph
* Word with its second highest TF-IDF in its first paragraph
....
* Word with its 4/d highest TF-IDF in its first paragraph

Figure 1. Defined Features



```
searchWord(List wordList, Feature featureItem){
    for each word in wordList
        if isMatch(word, featureItem)
            return word;
```

Figure 2. Feature Matching Analysis

procedure. Its status in the current text is generated for each word, and the status and the feature are compared with each other. The current word is generated as the attribute value in matching them. The status and the feature are viewed as the composite of the scope such as the entire text or its first paragraph, the relationship such as the frequency and the TF-IDF weight, and the rank.

The process of filling the string vector representing a text with words as feature values. We define the d features, $f_1, f_2, \ldots, f_d$, and view the process which is presented in Figure 2, as the function, $word_i = F(f_i, text)$, The string vector is filled with the words by applying the function as shown in equation (1),

$$\begin{aligned}\mathbf{str} &= [F(f_1, text), F(f_2, text), \ldots, F(f_d, text)] \\ &= [word_1, word_2, \ldots, word_d]\end{aligned} \tag{1}$$

The text is represented into an ordered finite set of words. The similarity between string vectors is computed based on the semantic similarity between two words.

We presented the three steps which are involved in encoding a text into a string vector. The difference of a string vector from a numerical vector is that its elements are given as strings. In the string vector as a text representation, the features are given as relations between words and the text and the feature values are given as words which correspond to the features. A string vector may be expanded into a string matrix by arranging more than one string vector as columns or rows. We need to define operations on string vectors for modifying machine learning algorithms into the versions which process them directly.

B. Similarity Metric

This section is concerned with the semantic similarity between two string vectors. In the previous section, we mentioned the process of mapping texts into string vectors. We need to define the semantic similarity metric between string vectors for modifying the KNN algorithm which is used for the text segmentation. We understand the semantic operation conceptually and start with defining the semantic similarity between words. This section is intended to describe the semantic similarity between two string vectors which represent paragraph pairs.

Let us mention the semantic operations conceptually for providing the background for defining the similarity metric

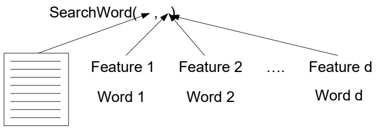


Figure 3. Word Assignment

$$\begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1N} \\ s_{21} & s_{22} & \cdots & s_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ s_{N1} & s_{N2} & \cdots & s_{NN} \end{bmatrix} \quad sim\left(t_i,t_j\right) = s_{ij} = \frac{2DF\left(t_i,t_j\right)}{DF\left(t_i\right) + DF\left(t_j\right)}$$

$$0 \le sim\left(t_i,t_j\right) \le 1.0$$

Figure 4. Similarity Matrix

between string vectors. They were initially proposed by Jo in 2015 [6] and defined as ones on strings based on their meaning under the assumption of each string with its own meaning. In [6], the semantic similarity between two strings, the semantic similarity average over strings, and the semantic similarity variance over them were defined as semantic operations and texts which are relevant to the word as its meaning. They were characterized mathematically and simulated on text collections with their variance domains. The first operation is adopted among the defined ones for defining the semantic similarity between string vectors.

The semantic similarity matrix between two words is illustrated in Figure 4. The two words are notated by t_i and t_j, the semantic similarity between them is done by $sim(t_i, t_j)$. $DF(t_i, t_j)$ is the number texts which include both words, t_i and t_j, $DF(t_i)$ and $DF(t_j)$ are the number of texts which include respectively, the word, t_i, and the word, t_j. The similarity between two texts is computed by equation (2),

$$sim(t_i, t_j) = \frac{2DF(t_i, t_j)}{DF(t_i) + DF(t_j)} \tag{2}$$

and is always given as a normalized value between zero and one. The rows and the columns of the matrix which is presented in Figure 4, correspond to the words in the corpus, and each element is given as a semantic similarity between two words.

A string vector is defined as an ordered finite set of strings as shown in equation (3),

$$\mathbf{str} = [str_1, str_2, ..., str_d] \tag{3}$$

The two string vectors are notated by equation (4) and (5),

$$\mathbf{str}_1 = [str_{11}, str_{12}, ..., str_{1d}] \tag{4}$$

$$\mathbf{str}_2 = [str_{21}, str_{22}, ..., str_{2d}] \tag{5}$$

The similarity between the two string vectors is defined as average over semantic similarities of one to one elements, as shown in equation (6),

$$sim(\mathbf{str}_1, \mathbf{str}_2) = \frac{1}{d}\sum_{i=1}^{d} sim(str_{1i}, str_{2i}) \tag{6}$$

The string vector which represents a text consists of words and the value of $sim(str_{1i}, str_{2i})$ is looked up from the similarity matrix which is presented in Figure 4. The similarity between the two string vectors, $\mathbf{str}_1$ and $\mathbf{str}_2$ is always given as a normalized value between zero and one.

We mentioned the similarity between two string vectors as a normalized value between zero and one. If the two string vectors are exactly same to each other as shown in equation (7),

$$\mathbf{str}_1 = \mathbf{str}_2 \tag{7}$$

the semantic similarity between them is 1.0 as shown in equation (8),

$$\begin{gathered} sim(\mathbf{str}_1, \mathbf{str}_2) = sim(\mathbf{str}_1, \mathbf{str}_1) = \\ \frac{1}{d}\sum_{i=1}^{d} sim(str_{1i}, str_{1i}) = 1.0 \end{gathered} \tag{8}$$

If the semantic similarities between elements of two string vectors are zeros, the sematic similarity between them is 0.0 as shown in equation (9),

$$sim(\mathbf{str}_1, \mathbf{str}_2) = \frac{1}{d}\sum_{i=1}^{d} sim(str_{1i}, str_{1i}) = \frac{0}{d} = 0.0 \tag{9}$$

Because $0 \leq sim(\mathbf{str}_1, \mathbf{str}_2) \leq 1$ the semantic similarity between them is always given as a normalized value between zero and one by equation (10),

$$\begin{gathered} 0 \leq sim(\mathbf{str}_1, \mathbf{str}_2) \leq 1 \\ 0 \leq \frac{1}{d}\sum_{i=1}^{d} sim(str_{1i}, str_{2i}) \leq 1 \end{gathered} \tag{10}$$

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

C. Proposed Version of KNN

This section is concerned with the proposed version of KNN algorithm which is presented in Figure 5, as the approach to the text segmentation. We mentioned the process of converting texts into string vectors, in Section III-A, and assume that the training examples and a novice one are given as string vectors. The semantic similarity between string vectors which is mentioned as the operation on them in Section III-B is used for selecting nearest neighbors from the training examples. A novice item is classified by voting ones of its nearest neighbors and variants may be derived by defining more voting schemes. This section is intended to describe the proposed version of KNN algorithm which classifies a string vector directly and its variants.

Let us mention the process of selecting the nearest neighbors for deciding the label of a novice item from the training examples. The sample paragraph pairs and the novice paragraph pair are encoded into string vectors by the process which is described in Section III-A. The similarities of a novice item with the sample ones by equation (6). The sample ones are ranked by their similarities and the most k similar ones are selected as the nearest neighbors. The rank based scheme is adopted in selecting the nearest neighbors in using the KNN algorithm to the classification task.

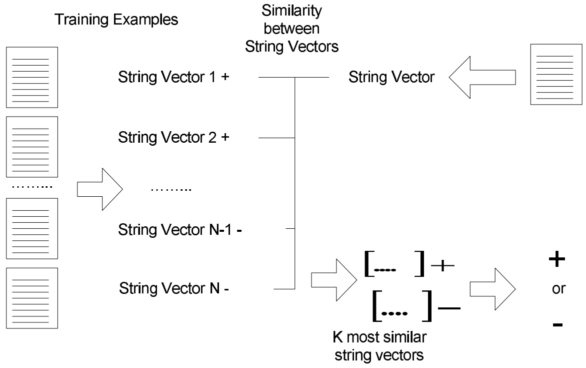


Figure 5. The Proposed Version of KNN

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, $\mathbf{str}$, whose elements are given as tables and their target labels, by equation (11),

$$\begin{aligned} Ne_k(\mathbf{str}) = \{(\mathbf{str}_1, y_1), (\mathbf{str}_2, y_2), \ldots, (\mathbf{str}_k, y_k)\}, \\ y_i \in \{c_1, c_2, \ldots, c_m\} \end{aligned} \tag{11}$$

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 $Count(Ne_k(\mathbf{str}), c_i)$. The label of the novice item, $\mathbf{str}$, is decided by the majority of categories in the nearest neighbors, as expressed by equation (12),

$$c_{\max} = \operatorname*{argmax}_{i=1}^{m} Count(Ne_k(\mathbf{str}), c_i) \tag{12}$$

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, $sim(\mathbf{str}, \mathbf{str}_1), sim(\mathbf{str}, \mathbf{str}_2), \ldots, sim(\mathbf{str}, \mathbf{str}_k)$ as weights, $w_1, w_2, \ldots, w_k$ by equation (13),

$$w_i = sim(\mathbf{str}, \mathbf{str}_i) \tag{13}$$

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 (14),

$$Weight(Ne_k(\mathbf{str}), c_i) = \sum_{\mathbf{str}_j \in c_i}^{k} w_j \tag{14}$$

The label of the novice item, $\mathbf{str}$, is decided by the category which corresponds to the maximum sum of weights as shown in equation (15),

$$c_{\max} = \operatorname*{argmax}_{i=1}^{m} Weight(Ne_k(\mathbf{str}), c_i) \tag{15}$$

When the weights of nearest neighbors are set constantly, equation (15) is same to equation (12), as expressed in equation (16),

$$Weight(Ne_k(\mathbf{str}), c_i) = Count(Ne_k(\mathbf{str}), c_i) \tag{16}$$

We described the proposed version of the KNN algorithm in this section. In using the proposed KNN algorithm, raw data is encoded into string vectors, 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. Application to Text Segmentation

This section is concerned with the text segmentation system which adopts the string vector based KNN algorithm. In Section III-C, we described the proposed version of KNN algorithm as the approach to the text segmentation. It was viewed into the binary classification of each adjacent paragraph pair into boundary or continuance. A text is subtexts based on its contents by putting the boundary between paragraphs in the pair which is classified into boundary. This section is intended to describe the text segmentation system with respect to its functions and architecture.

The sample paragraph pairs which are labeled with boundary or continuance are illustrated in Figure 6. The text segmentation is viewed into the binary classification where each adjacent paragraph is classified into one of the two categories. A paragraph pair which is labeled with boundary means the topic transition between the two paragraphs, and one labeled with continuance does the topic continuation between them. The text segmentation belongs to the domain dependent task where each item is classified differently depending on the domain. Before executing the text segmentation, the input text domain should be presented.

The entire architecture of the proposed text segmentation system is illustrated in Figure 7. A text is given as the input and adjacent paragraph pairs are extracted by partitioning a text into paragraphs and slide them with the two sized window. The sample paragraph pairs in the boundary group and the continuation group, and ones which extracted from the text are mapped into string vectors in the encoding module. The paragraph pairs from the text are classified into one of the two categories in the similarity computation module and the voting module. In the input text, the boundary is put between paragraphs in each pair which is classified into boundary and the text is partitioned into subtexts by the boundaries.

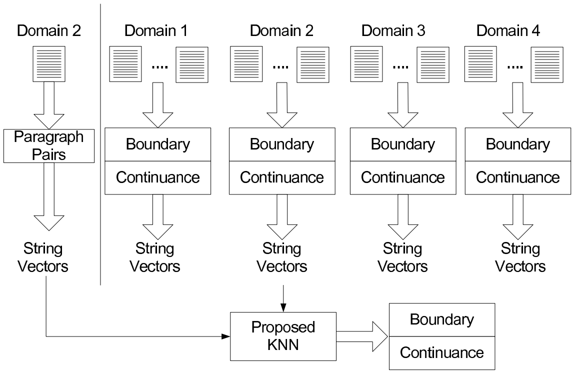


Figure 6. Sample Paragraph Pairs

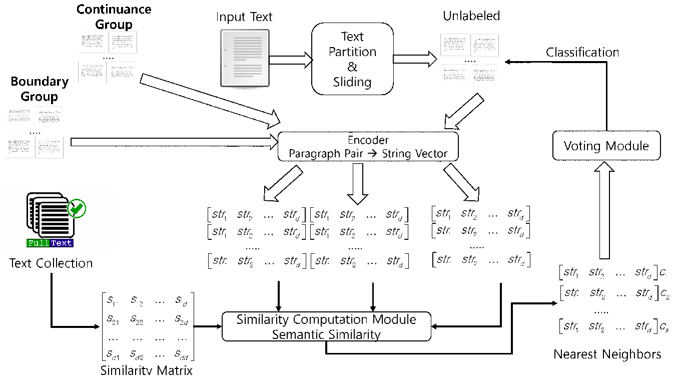


Figure 7. Proposed System Architecture

The execution process of the proposed system is illustrated as a block diagram in Figure 8. The sample paragraph pairs which are labeled with boundary or continuance are collected from the domain, and encoded into string vectors. Adjacent paragraph pairs are extracted from the input text by partitioning it into paragraphs and sliding them with the two sized window, and also encoded into string vectors. The nearest neighbors are selected by the similarity computation, its label is decided by voting ones of its nearest neighbors for each paragraph. The partitions for generating subtexts is are put between adjacent paragraph pairs which are classified into boundary.

Let us make some remarks on the proposed system which is illustrated in Figure 7 as the architecture. The text segmentation is defined as the binary classification where each adjacent paragraph pair is classified into boundary or continuance. Each paragraph pair is encoded into a string vector, instead of a numerical vector, and a string vector is classified directly. The input text is partitioned by the bound-

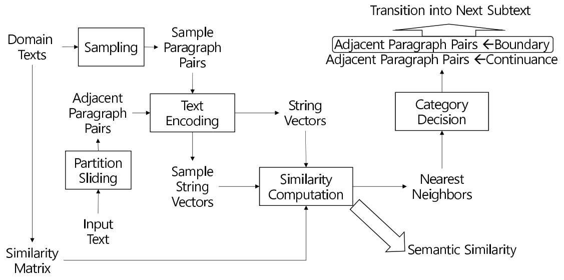


Figure 8. Execution Process of Proposed System

ary which is put between paragraphs in each pair which is classified into boundary. Subtexts as the content based divisions of the input text may be treated as independent ones.

IV. Experiments

This section is concerned with the empirical experiments for validating the proposed version of KNN, and consists of the five sections. In Section IV-A, we present the results from applying the proposed version of KNN to the text segmentation on the collection, NewsPage.com. In Section IV-B, we show the results from applying it for classifying paragraph pairs into boundary or continuance, from the collection, Opinosis. In Section IV-C and IV-D, we mention the results from comparing the two versions of KNN with each other in the task of text segmentation from 20NewsGroups.

A. *NewsPage.com*

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

In Table I, we specify the text collection, NewsPage.com, which is used in this set of experiments. The collection was used for evaluating approaches to text categorization tasks in previous works [**?**]. In each category, we extract 250 adjacent paragraph pairs and label them with boundary or continuance, keeping the complete balance over the two labels. In each category, the set of 250 paragraph pairs is partitioned into the training set of 200 ones and the test set of 50 ones. Each text is segmented into paragraphs by a carriage return, and adjacent paragraph pairs are generated by sliding two sized window on the list of paragraphs.

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

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

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

In Figure 9, we illustrate the experimental results from classifying each adjacent paragraph pair into boundary or continuance, using the both versions of KNN algorithm. The y-axis indicates the accuracy which is the rate of the correctly classified examples in the test set. Each group in the x-axis means the domain within which the text summarization which is viewed as a binary classification is performed, independently. In each group, the gray bar and the black bar indicate the accuracies of the traditional version and the proposed version of the KNN algorithm. The most right group in Figure 9 consists of the averages over the accuracies of the left four groups, and the input size which is the dimension of numerical vectors is set to 50.

Let us make the discussions on the results from doing the text segmentation, using the both versions of KNN algorithm, as shown in Figure 9. The accuracy which is the performance measure of this classification task is in the range between 0.4 and 0.64. The proposed version of KNN algorithm works strongly better in the all domains. The both versions work best in the domain, Business, in the comparison of domains. From this set of experiments, we conclude the proposed version works better than traditional one, in averaging over the four cases.

B. *Opinopsis*

This section is concerned with the set of experiments for validating the better performance of the proposed version

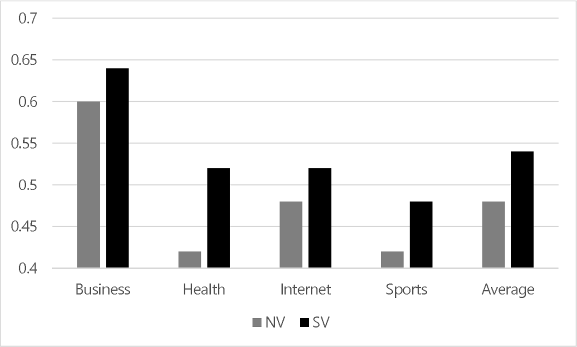


Figure 9. Results from Segmenting Texts in Text Collection: NewsPage.com

on the collection, Opinosis. We view the text segmentation into a binary classification where each adjacent paragraph pair is classified into boundary or continuance, and collect the paragraphs pairs, sliding paragraphs in each text by two sized window and labeling manually with one of boundary and continuance from the collection. Each paragraph pair is exclusively classified into one of the two labels. We fix the input size to 50 and use the accuracy as the evaluation measure. In this section, we observe the performance of the both versions of KNN algorithm, in the three experiments as many as topics.

In Table II, we specify the text collection, Opinosis, which is used in this set of experiments. The test collection is used in previous works for evaluating approaches to text categorization. We extract the 50 adjacent paragraph pairs in each topic, and label them with ‘boundary’ or ‘continuance’, keeping the complete balance. The set of 50 paragraph pairs is portioned into the 40 as the training set and the 10 as the test set, in each topic. In the process of generating the paragraph pairs, each text is segmented into paragraphs by the carriage return, the adjacent paragraph pairs are generated by sliding the paragraphs.

Table II
THE NUMBER OF TEXTS AND PARAGRAPH PAIRS IN OPINIOPSIS

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

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

In Figure 10, we illustrate the experimental results from the text segmentation which is mapped into a classification task, using the both versions of KNN algorithm. Like Figure 9, the y-axis indicates the value of accuracy, and the x-axis indicates the group of two versions by a domain of Opniopsis. In each group, the gray bar and the black bar indicate the results of the traditional version and the proposed version of KNN algorithm. In Figure 10, the most right group indicates the averages of the both version over their results of the left three groups. Therefore, Figure 10 shows the results from classifying adjacent paragraph pairs into one of ‘boundary’, and ‘continuance’, by the both versions.

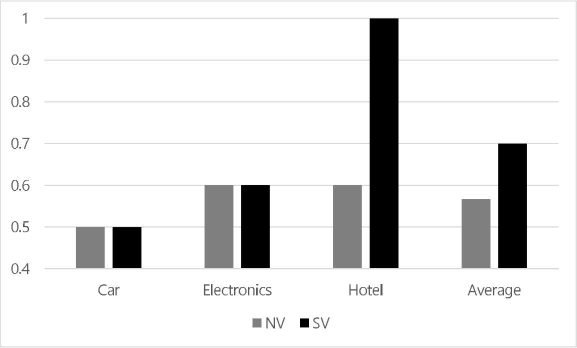


Figure 10. Results from Segmenting Texts in Text Collection: Opiniopsis

We discuss the results from doing the text segmentation which is mapped into a binary classification, using the both versions of KNN algorithm, shown in Figure 10. The accuracy values of the both versions range between 0.5 and closely to 1.0. The proposed version shows its perfect performance in the domain, Hotel. It is comparable with the traditional version in the others. From this set of experiments, we conclude that the proposed one works better in averaging the three cases.

C. 20NewsGroups I: General Version

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

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

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

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

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

In Figure 11, we illustrate the experimental results from deciding whether we put a boundary, or not, between two adjacent paragraphs, on the broad version of 20NewsGroups. Figure 11 has the identical frame of presenting the results to those of Figure 9 and 10. In each group, the gray bar and the black bar indicates the achievements of the traditional version and the proposed version of KNN algorithm, respectively. In the x-axis, each group indicates the domain within which each paragraph pair is classified into 'boundary', or 'continuance'. This set of experiments consists of the four binary classifications in each of which it is done so.

Let us discuss the results from doing the text segmentation using the both versions of KNN algorithm as shown in Figure 11. The accuracies of both versions range between 0.45 and 0.65. The proposed version shows its better performances in three of the four domains; it shows its outstanding difference from the traditional version in the domain, sci. However, its performance is leaded in the domain, rec. From this set of experiments, the proposed version wins over the traditional one, in averaging its achievements of the four domains.

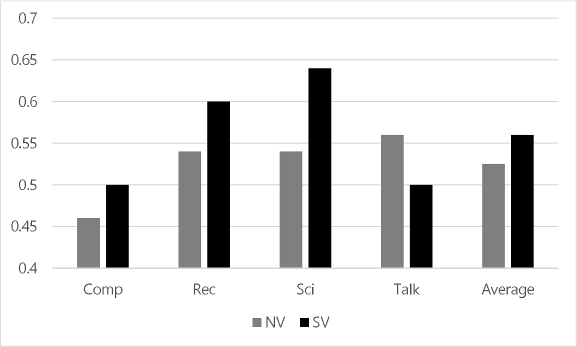


Figure 11. Results from Segmenting Texts in Text Collection: 20NewsGroup I

D. 20NewsGroups II: Specific Version

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

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

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

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

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

We present the experimental results from classifying the paragraph pairs using the both versions of KNN algorithm on the specific version of 20NewsGroups. The frame of illustrating the classification results is identical to the previous ones. In each group, the gray bar and the black bar stand for the achievements of the traditional version and the proposed version, respectively. The y-axis in Figure 12, indicates the classification accuracy which is used as the performance metric. In this set of experiments, we execute the four independent classification tasks which correspond to their own domains, where each paragraph pair is classified into 'boundary' or 'continuance'.

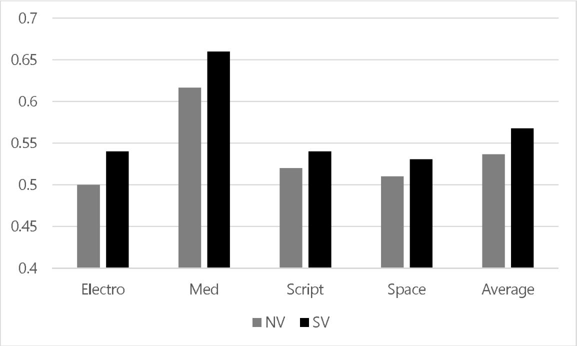


Figure 12. Results from Segmenting Texts in Text Collection: 20News-Group II

Let us discuss the results from classifying the adjacent paragraph pairs using the both versions of KNN algorithm on the specific version of 20NewsGroups, as shown in Figure 12. The accuracies as the performance metrics of this classification task which is mapped from the text segmentation range between 0.45 and 0.66. The proposed version shows its better results in all of the four domains. The differences between the both versions are almost same to each other in the four domains. From this set of experiments, it is concluded that the proposed version have its better performance by averaging over the accuracies of the four domains.

V. Conclusion

Let us discuss the results from segmenting a text using the two versions of KNN algorithm. In these sets of experiments, we compare the two versions with each other in the classification tasks which is mapped from the text segmentations. The proposed version shows its better results in all of the four collections. The classification accuracies of the traditional version range between 0.41 and 0.62, while those of the proposed version range between 0.44 and 0.88. From the four sets of experiments, we conclude that the proposed version improves the text segmentation performance, as the contribution of this research.

We need to consider the remaining tasks for doing the further research. We will apply and validate the proposed approach in segmenting a text in the specific domains such as medicine, engineering, and law, rather than the general domains. In order to improve the performance, we may consider various types of features of string vectors. As another scheme of improving the performance, we define and combine multiple similarity measures between two string vectors with each other. By adopting the proposed approach, we may implement the text segmentation system as a module or an independent system.

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] 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.

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

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

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

[6] T. Jo, "Simulation of Numerical Semantic Operations on String in Text Collection", 45585-45591, International Journal of Applied Engineering Research, Vol 10, No 24, 2015.

[7] 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.

[8] 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.

[9] T. Jo, "Summarizing News Articles by Feature Similarity based Version of K Nearest Neighbor", 51-52, The Proceedings of 1st International Conference on Advanced Engineering and ICT-Convergence, 2018.

[10] T. Jo, “Using Table based Version of K Nearest Neighbor for Segmenting News Articles by their Contents”, 62-64, The Proceedings of 1st International Conference on Advanced Engineering and ICT-Convergence, 2018.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] T. Jo, “K Nearest Neighbor for Text Categorization using Feature Similarity”, 99-104, The Proceedings of 2nd International Conference on Advanced Engineering and ICT Convergence, 2019.

[17] 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.

[18] T. Jo, “Automatic Summarization System in Current Affair Domain by Table based K Nearest Neighbor”, 115-121, The Proceedings of 2nd International Conference on Advanced Engineering and ICT-Convergence, 2019.

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

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

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

[22] Taeho Jo, “Applying Table based AHC Algorithm to News Article Clustering”, pp 8-11, The Proceedings of International Conference on Green and Human Information Technology, Part I, 2019.

[23] 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.

[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”, pp913-920, Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, 2006.

[27] A. Karatzoglou and I. Feinerer, “Text Clustering with String Kernels in R”, pp91-98, Advances in Data Analysis, 2006.

[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] 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.