String Vector based AHC Algorithm for Clustering Words Semantically

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

***Abstract*—This article proposes the modified AHC (Agglomerative Hierarchical Clustering) algorithm which clusters string vectors, instead of numerical vectors, as the approach to the word clustering. The results from applying the string vector based algorithms to the text clustering were successful in previous works and synergy effect between the text clustering and the word clustering is expected by combining them with each other; the two facts become motivations for this research. In this research, we define the operation on string vectors called semantic similarity, and modify the AHC algorithm by adopting the proposed similarity metric as the approach to the word clustering. The proposed AHC algorithm is empirically validated as the better approach in clustering words in news articles and opinions. We need to define and characterize mathematically more operations on string vectors for modifying more advanced machine learning algorithms.**

***Keywords*-Word Clustering; Agglomerative Hierarchical Clustering Algorithm; String Vector**

I. Introduction

Word clustering refers to the process of segmenting a group of various words into subgroups of content based similar words. In the task, a group of arbitrary words is given as the input, and they are encoded into their structured forms. The similarity measure between the structured forms which represent the words is defined and the similarities among them are computed. The words are arranged into subgroups based on their similarities. In this research, we assume that the unsupervised learning algorithms are used for the task, although other types of approaches exist.

Let us mention the challenges which this research attempts to tackle with. In encoding words into numerical vectors for using the traditional clustering algorithms, we need many features for the robust clustering, since each feature has very weak coverage[27]. Each numerical vector which represents a word or a text tends to have the sparse distribution where zero values are dominant with more than 90%[22]. In previous works, we proposed that texts or words should be encoded into tables as alternative representations to numerical vectors, but it is very expensive to compute the similarity between tables[22]. Hence, in this research, we solve the problems by encoding words into string vectors.

Let us mention what is proposed in this research as its idea. We encode words into string vectors where elements are text identifiers which are given as symbols or codes as alternative representations to numerical vectors. We define the operation on string vectors which corresponds to the cosine similarity between numerical vectors as the similarity measure between them. We modify the AHC (Agglomerative Hierarchical Clustering) algorithm into the version where input data are given as string vectors. Hence, in this research, the words are clustered by the modified version of AHC algorithm.

Let us consider the benefits which are expected from this research. The string vectors become the more compact representations of words than numerical vectors; it requires much less features in encoding words. We expect the improved discriminations among string vectors since there is very few sparse distributions in string vectors. We expect the improved clustering performance by solving the problems which are caused by encoding words into numerical vectors. Therefore, the goal of this research is to implement the word clustering systems which are improved by the benefits.

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 significance 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 explore the previous cases of applying the AHC 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. Using AHC Algorithm to Text Mining Tasks

This section is concerned with the previous cases of using the modified version of AHC algorithm for the word clustering and the text clustering. In previous works, the AHC algorithm was modified and applied to the word

clustering as a supervised learning algorithm. We will survey cases of applying it to the text clustering, as well as the word clustering. The scope of this section is restricted to the process of clustering words based on their meanings. This section is intended to mention the previous cases of applying the modern versions of the AHC algorithm and the KNN algorithm to classification tasks and clustering tasks.

Let us explore the cases of using the modernized AHC algorithm for the word clustering tasks. The similarities among features were considered in using the AHC algorithm for the word clustering, in order to avoid the poor discriminations among sparse vectors [4]. The AHC algorithm which clusters tables directly was proposed as the approach to the word clustering [7]. The AHC algorithm was modified into the version which clusters graphs as its modernization in the word clustering [8]. In the above literatures, the modernized AHC algorithms with their various directions were proposed as the approaches to the word clustering.

The text clustering may be considered as another type of data clustering. The AHC algorithm which uses the similarity metric which considers the similarities among attributes was applied to text clustering [12]. The modernized AHC algorithm which clusters table directly was mentioned as the approach to the text clustering [11]. Another modernized AHC algorithm which clusters graphs directly was adopted for implementing the text clustering system [19]. The text which is expanded from a word consists of more than paragraph.

The clustering index will be used as the evaluation metric of clustering results. It was initially mentioned for evaluating the dynamic data organization in 2006 [1]. It was described as the evaluation metric of clustering results in 2007 [24]. It was used for tuning parameters of clustering algorithms [20]. The clustering index which was mentioned in the above literatures is the integrated one of the intra-cluster similarity and the inter-cluster discrimination, following style in the F1 measure.

Let us mention some points which are distinguished from the above literatures. We surveyed the previous works on applying the modernized versions of AHC algorithm to the word clustering and the text clustering. We presented the historical review of proposing and using the clustering index which is used as the evaluation metric in this study. The proposed version of AHC algorithm as the approach to the word clustering clusters directly string vectors which represents words. In this study, we encode words into string vectors, and clusters them using the proposed version of AHC algorithm.

B. Word and Text Encoding

This section is concerned with the previous cases of encoding words or texts into non-numerical vectors. The problems in encoding them into numerical vectors have been mentioned several times, until now. Previously, previous works challenged against the problems by encoding them into alternative structured data to numerical vectors. We mention the tables, the string vectors, and the graphs as alternative structured types. This section is intended to survey the previous cases of encoding words and texts into non-numerical vectors in other tasks.

Let us mention the previous cases of encoding texts or words into tables in text mining tasks. Words were encoded into tables in using the KNN algorithm for the word categorization [13]. Words were encoded so in using it for the keyword extraction [14]. Texts were encoded into tables in using it for the text categorization [18]. In the above literatures, texts or words were encoded into tables in using the KNN algorithm.

Let us mention the cases of encoding texts or words into string vectors. It was proposed that words should be encoded into string vectors, in using the KNN algorithm for the word categorization [18]. Encoding words into string vectors was also proposed in using the KNN algorithm for extracting keywords [16]. It was proposed that texts should be encoded into string vectors, in using the KNN algorithm for the text categorization [17]. The above literatures presented the previous cases of encoding raw data into string vectors.

Let us consider the previous works on encoding texts or words into graphs. Words were encoded into graphs in using the KNN algorithm for the word categorization [9]. In using it for the keyword extraction, words were encoded so [10]. Texts were encoded into graphs in using it for the text categorization [21]. In the above literatures, we presented the previous cases of encoding raw data into graphs.

We mentioned the three schemes of encoding words or texts in the previous works. We adopt the second scheme where words are encoded into string vectors. We define the similarity metric between two string vectors and use it for modifying the AHC algorithm into the version which clusters string vectors directly. We use the modified version of AHC algorithm for implementing the word clustering system. We validate the modified AHC algorithm empirically by comparing it with the traditional version, in clustering words.

C. Non-Numerical Vector based Clustering Algorithms

This section is concerned with the previous works on the clustering algorithms which process non-numerical vectors. In the previous section, we presented the cases of encoding texts or words into tables, string vectors, or graphs, to avoid issues in encoding them into numerical vectors. In this section, we mention the string kernel based clustering algorithms, the table matching algorithm, and the Neural Text Self Organizer, as the non-numerical vector unsupervised learning algorithms for clustering texts. Because the text clustering belongs to the clustering task, together with the word clustering, in this section, we focus on the text clustering, in surveying the previous works. This section is

intended to survey the previous works about this type of clustering algorithms as the approaches to the text clustering.

Let us consider the clustering algorithms with the string kernel which used for clustering texts. The string kernel was initially proposed for improving the performance of the SVM (Support Vector Machine) as the approach to the text classification by Lodhi et al. in 2002 [26]. It was used for modifying the k means algorithm, and the algorithm was implemented in R by Karatzoglou and Feinerer in 2006 [25]. The spectral algorithm was modified using the string kernel and validated in the text clustering empirically by Shi et al. in 2010 [28]. The string kernel which is the kernel function of raw texts was used for clustering texts in the above literatures.

Let us survey the previous works on the table based matching algorithm. It was initially proposed as the approach to the text categorization by Jo and Cho in 2008 [22]. It was utilized to the soft classification of texts which allows to assign more than one category to each text by Jo in 2008 [2]. It was improved into the more robust and stable approach to the text categorization by Jo in 2015 [5]. In the approach which is mentioned in the above literatures, texts should be encoded into tables.

Let us mention the Neural Text Self Organizer as the unsupervised neural network model which is specialized for the text clustering. It was initially proposed as the approach to the text clustering by Jo and Japkowicz in 2005 [23]. It was mentioned as a main method of text clustering by Zheng et al. in 2006 [29]. Its performance was empirically validated in clustering texts in various domains by comparing it with the k means algorithm and the Kohonen Networks [3]. Texts should encoded into string vectors as non-numerical vectors, in using it.

We mentioned above the non-numerical vector based clustering algorithms and classification algorithm. The string kernel based clustering algorithm processes raw texts directly, the table based machine algorithm classifies tables directly, and the Neural Text Self Organizer clusters string vectors, directly. In this research, words are encoded into string vectors. The AHC algorithm is modified into the version which clusters string vectors directly as the approach to the word clustering. Its clustering performance will be validated by comparing with the traditional AHC algorithm in the semantic word clustering.

III. Proposed Approach

This section is concerned with encoding words into string vectors, modifying the AHC (Agglomerative Hierarchical Clustering) algorithm into string vector based version and applying it to the word clustering, and consists of the three sections. In Section III-A, we deal with the process of encoding words 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 AHC version as the approach to the word clustering, and in Section III-D, present the architecture of the system which we try to implement by adopting the proposed AHC algorithm. Therefore, this article is intended to describe the proposed AHC version as the word clustering tool.

A. Word Encoding

This section is concerned with the process of encoding words into string vectors. We presented the previous cases of encoding texts into string vectors in Section II-B and II-C. This process involves the three steps: feature definition, feature matching analysis, and text identifier assignment. A string vector which represents a word consists of text identifiers which are related with the word as the elements. This section is intended to describe the three steps which are presented in Figure 1-3.

* Text where word have its first highest frequency in the entire
* Text where word have its second highest frequency in the entire
.....
* Text where word have its 4/d highest frequency in the entire

* Text where word have its first highest TF-IDF weight in the entire
* Text where word have its second highest TF-IDF weight in the entire
.....
* Text where word have its 4/d highest TF-IDF weight in the entire

* Text where word have its first highest frequency in its first paragraph
* Text where word have its second highest frequency in its first paragraph
.....
* Text where word have its 4/d highest frequency in its first paragraph

* Text where word have its first highest TF-IDF in its first paragraph
* Text where word have its second highest TF-IDF in its first paragraph
.....
* Text where word have its 4/d highest TF-IDF in its first paragraph

Figure 1. Defined Features

The features which are defined in encoding words into string vectors are illustrated in Figure 1. The assumption which are inherent in defining the features is that the first paragraph in each text is its key part and the dimension of each string vectors is d. The frequency and the TF-IDF

(Term Frequency and Inverse Document Frequency) weight are relationships between a word and a text for defining the features. The group of features is divided into the four subgroups which consist of $d/4$ features and texts are ranked by the frequency and TF-IDF weight in the entire text or its first paragraph within the subgroup. The process of defining the features which are presented in Figure 1, is characterized as manual and arbitrary one by a system developer or a system administrator.

```
searchTextID(List textIDList, Feature featureItem, Word wordItem){
    for each textID in textIDList
        if isMatch(textID, featureItem, wordItem)
            return textID;
```

Figure 2. Feature Matching Analysis

The process of feature matching analysis is illustrated as a pseudo code in Figure 2. The list of texts in the corpus, feature which is one among what is presented in Figure 1, and the word are given as arguments. For each text, the relationship of word is extracted and whether the current relationship and the feature which is given as an argument match with each other is checked. If they match, the current text is returned. The three parts of each feature in the implementation level exist: the frequency or the weight, the scope which is the entire text or the first paragraph, and the rank.

The process of assigning a text to each feature of the string vector which represents a word is illustrated in Figure 3.

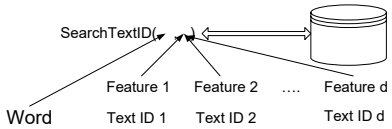

Figure 3. Text Identifier Assignment

We define d features which are notated by $f_1, f_2, \ldots, f_d$, as illustrated in Figure 1, and define the process which illustrated in Figure 2, as the function of the word and the features:$text\_id_i = F(f_i, word)$. The string vector which represents a word is filled with text identifiers which correspond to their own features as follows:

$$\mathbf{str} = [F(f_1, word), F(f_2, word), \ldots, F(f_d, word)]$$

The string vector is viewed as an ordered finite set of text identifiers, as follows:

$$\mathbf{str} = [text\_id_1, text\_id_2, \ldots, text\_id_d]$$

A numerical values is replaced by a string in each position, comparing a numerical vector with a string vector.

In this section, we described the three steps which are presented in Figure 1-3, for encoding a word into a string vector. The difference of a string vector from a numerical vector is that elements are given as strings instead of numerical values. In the string vector which represents a word, the features are given as relationships between a word and a text, and the feature values are given as identifiers of texts which correspond to the features of a word. A string vector may

be expanded into a string matrix which consists of strings with the two axis. We need to define the operations on string vectors for modifying the machine learning algorithms into versions which process them directly.

B. Semantic Similarity between String Vectors

This section is concerned with the semantic similarity between two string vectors. In the previous section, we mention the process of encoding words into string vectors. We need to define the similarity metric between string vectors, for modifying the AHC algorithm into the version which clusters string vectors directly. In order to do that, we introduce the concept of semantic operations, and define the semantic similarity between text identifiers as a semantic operations. This section is intended to describe the semantic similarity between two string vectors.

The semantic operations on strings are ones based on meanings of strings under the assumption of each string with its own meaning. They were initially proposed as the basis for dealing with strings by Jo in 2015 [6]. The semantic similarity between two strings, the semantic similarity average, and the semantic similarity variance were defined as typical semantic operations. They were characterized mathematically and simulated in text collections with their various domains. We adopt the first operation, semantic similarity, for modifying the AHC algorithm as the approach to the word clustering.

In Figure 4, the semantic similarity matrix between two texts is illustrated. The two texts are notated by d_i and d_j and the similarity between them is notated by $sim(d_i, d_j)$. The two texts, d_i and d_j, are expressed as the two sets of words, D_i and D_j, and $|D_i|$ and $|D_j|$ are cardinalities of the two sets. The similarity between two texts is computed by equation (1),

$$sim(d_i, d_j) = \frac{2|D_i \cap D_j|}{|D_i| + |D_j|} \tag{1}$$

and the similarity 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 texts in the corpus, and each element becomes the similarity between corresponding texts.

A string vector is defined as an ordered finite set of strings as shown in equation (2),

$$\mathbf{str} = [str_1, str_2, ..., str_d] \tag{2}$$

The two string vectors are notated by equation (3) and (4),

$$\mathbf{str}_1 = [str_{11}, str_{12}, ..., str_{1d}] \tag{3}$$

$$\mathbf{str}_2 = [str_{21}, str_{22}, ..., str_{2d}] \tag{4}$$

The similarity between the two string vectors is defined as average over semantic similarities of one to one elements,

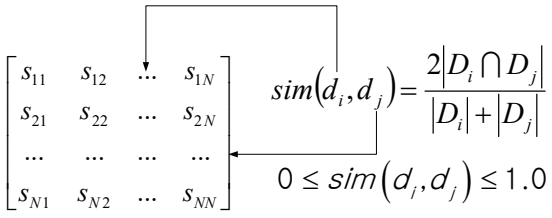

Figure 4. Similarity Matrix

as shown in equation (5),

$$sim(\mathbf{str}_1, \mathbf{str}_2) = \frac{1}{d} \sum_{i=1}^{d} sim(str_{1i}, str_{2i}) \tag{5}$$

The string vector which represents a word consists of text identifiers 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 (6),

$$\mathbf{str}_1 = \mathbf{str}_2 \tag{6}$$

the semantic similarity between them is 1.0 as shown in equation (7),

$$sim(\mathbf{str}_1, \mathbf{str}_2) = sim(\mathbf{str}_1, \mathbf{str}_1) = \frac{1}{d} \sum_{i=1}^{d} sim(str_{1i}, str_{1i}) = 1.0 \tag{7}$$

If the semantic similarities between elements of two string vectors are zeros, the sematic similarity between them is 0.0 as shown in equation (8),

$$sim(\mathbf{str}_1, \mathbf{str}_2) = \frac{1}{d}\sum_{i=1}^{d} sim(str_{1i}, str_{1i}) = \frac{0}{d} = 0.0 \quad (8)$$

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 (9),

$$\begin{aligned} 0 \leq sim(\mathbf{str}_1, \mathbf{str}_2) \leq 1 \\ 0 \leq \frac{1}{d}\sum_{i=1}^{d} sim(str_{1i}, str_{2i}) \leq 1 \end{aligned} \quad (9)$$

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

C. The Proposed Version of AHC Algorithm

This section is concerned with the proposed version of AHC algorithm which is shown in Figure 5, as the approach to the semantic word clustering. We described the process of encoding words into string vectors in Section III-A, and assume that items in the group are given as string vectors. The semantic similarity metric between two string vectors which were described in Section III-B is used for proceeding clustering items by the AHC algorithm. More variants may be derived from it by proposing more schemes of computing the cluster similarity and merging clusters. This section is intended to describe the proposed version of the AHC algorithm which clusters string vectors, directly, and its variants.

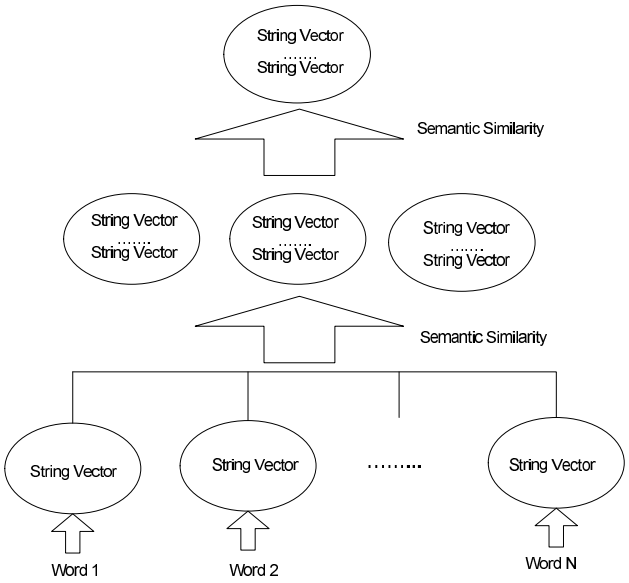

Figure 5. The Proposed Version of AHC Algorithm

Let us mention the computation of the similarity between two clusters. The two clusters are notated by sets of string vectors: $C_1 = \{\mathbf{str}_{11}, \mathbf{str}_{12}, \ldots, \mathbf{str}_{1|C_1|}\}$ and $C_2 = \{\mathbf{str}_{21}, \mathbf{str}_{22}, \ldots, \mathbf{str}_{2|C_2|}\}$. All possible pairs are generated from the two clusters, and for each pair, its similarity is computed by the equation which was defined in Section 3.2. The similarity between the two clusters is computed by equation (10),

$$sim(C_1, C2) = \frac{1}{|C_1||C_2|}\sum_{i=1}^{|C_1|}\sum_{j=1}^{|C_2|} sim(\mathbf{str}_{1i}, \mathbf{str}_{2j}) \quad (10)$$

The similarity between two string vectors is always given as a normalized value between zero and one, so the similarity between two clusters which is computed by equation (10) is also given as a normalized value.

Let us mention the process of clustering data items by the AHC algorithm. The tables which are mapped from words in the group are notated by the set, $\{\mathbf{str}_1, \mathbf{str}_2, \ldots, \mathbf{str}_N\}$, and the set of initial clusters is expressed as $\{C_1^1, C_2^1, \ldots, C_{N_1}^1\}$, where $C_i = \{\mathbf{str}_i\}$, the super script 1 means the initial iteration, and $N_1 = N$ which is the number of clusters in the first iteration. All possible pairs of clusters, $Pair(C_i^k, C_j^k), i < j$, are generated, and the similarity between two clusters $sim(C_i^k, C_j^k)$ is computed for each pair by equation (10). Clusters in the pair with the maximal similarity are merged into a cluster as shown in equation (11),

$$\begin{aligned} Pair_{\max}(C_i^k, C_j^k) &= \underset{i<j}{\overset{N_k}{\operatorname{argmax}}}\, sim(C_i^k, C_j^k) \\ C_{merge}^{k+1} &= merge(Pair_{\max}(C_i^k, C_j^k)) \end{aligned} \quad (11)$$

and the number of clusters in the $k+1$ th iteration is $N_{k+1} = N_k - 1$ by decrementing the number of clusters by merging it. The AHC algorithm proceeds clustering by iterating the computation of similarities between clusters in all possible pairs and merge of pair with the maximal similarity into one cluster.

Let us mention the clustering index which is used for evaluating the traditional version and the proposed one of the AHC algorithm. The intra-cluster similarity of the cluster, C_i, and the inter-cluster similarity of the two clusters, C_i and C_j are notated respectively by $intra\_sim(C_i)$ and $inter\_sim(C_i, C_j)$ and the clustering results are expressed as a set of clusters, $C = \{C_1, C_2, \ldots, C_{|C|}\}$. The intra-cluster similarity over the clustering results,C, is computed by equation (12),

$$intra\_sim(C) = \frac{1}{|C|}\sum_{i=1}^{|C|} intra\_sim(C_i) \quad (12)$$

and the inter-cluster similarity over entire cluster, C is computed by equation (13),

$$inter\_sim(C) = \frac{2}{|C|(|C|-1)}\sum_{i<j}^{|C|} inter\_sim(C_i, C_j) \quad (13)$$

The clustering index is computed by equation (14),

$$CI(C) = \frac{2 \cdot intra\_sim(C) \cdot (1 - inter\_sim(C))}{intra\_sim(C) + (1 - inter\_sim(C))} \quad (14)$$

The desired goal of clustering data items is to maximize the intra cluster similarity and minimize the inter cluster similarity.

We described the proposed version of the AHC algorithm as the approach to the data clustering. Raw data is encoded into string vectors for using the proposed version for clustering data items. We use the similarity metric between string vectors for computing similarities among items. The similarity between clusters is the average over all possible similarities of data items. The desired number of clusters is set as the termination condition in proceeding clustering by the AHC algorithm.

D. Word Clustering System

This section is concerned with the semantic word clustering system which adopts the string vector based AHC algorithm. We described the proposed version of the AHC algorithm as the approach to the semantic word clustering in Section III-C. Words in the group are encoded into string vectors and clustered into subgroups in the system. Clustering data items is executed by iterating computing similarities among clusters and merging clusters. This section is intended to describe the semantic word clustering system with respect to its functions and architecture.

The words are gathered as clustering targets. Because unsupervised learning algorithms are used for clustering data, the words are assumed to be unlabeled. The words are encoded into tables by the process which was mentioned in Section III-A. The similarity metric which is described in Section III-B is defined and the AHC algorithm which is described in Section III-C is adopted as the clustering method. The number of clusters should be set as the termination condition in the system.

The entire architecture of the proposed word clustering system is illustrated in Figure 6. All words which are given as the input are encoded into string vectors. They are clustered by the AHC algorithm which was described in Section III-C in the similarity computation module and the clustering module. The string vector clusters are restored into the word clusters by the decoder. There are the four modules in the system: the encoding module, the similarity computation module, the clustering module, and the decoding module.

The execution process of the proposed system is illustrated as a block diagram in Figure 7. The words which are clustered are encoded into string vectors by the encoding module. The string vectors are clustered by the AHC algorithm by iterating computing the similarity among clusters and merging clusters. Clusters each of which contain semantic similar words are given as the final output in the

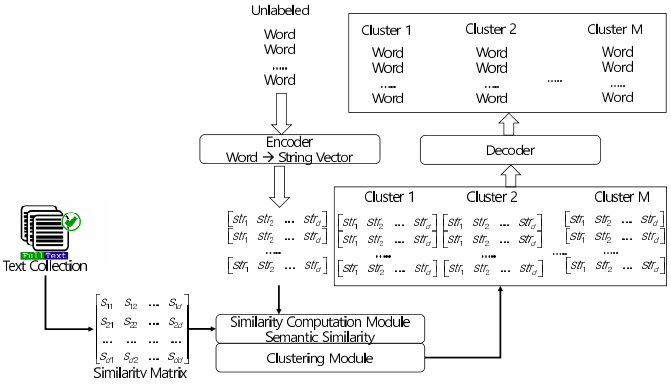

Figure 6. Proposed System Architecture

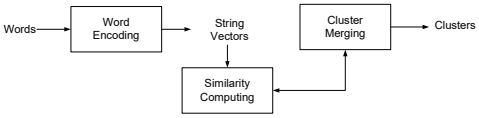

Figure 7. Execution Process of Proposed System

system. In advance we need to decide the number of clusters as an external parameter.

Let us make some remarks on the proposed system which is illustrated in Figure 6 as the architecture. Words are encoded into string vectors, instead of numerical vectors. String vectors which represent words are clustered by the proposed AHC algorithm, directly. The clustering performance is improve by what is proposed in this research as shown in Section IV. In the next research, we present the graphical user interface and the source codes 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 AHC algorithm, and

consists of the five sections. In Section IV-A, we present the results from applying the proposed version of AHC to the word clustering on the collection, NewsPage.com. In Section IV-B, we show the results from applying it for clustering words from the collection, Opinosis. In Section IV-C and IV-D, we mention the results from comparing the two versions of AHC algorithm with each other in clustering 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. We set the number of clusters as four, following the number of categories for evaluating the performance, and gather words from the collection, category by category, as the labeled ones. In the clustering process, each word is arranged into one of the four clusters, exclusively, in this set of experiments. We use the clustering index which was proposed in [1] for evaluating the clustering performances. Therefore, this section is intended to observe the performance of the traditional and proposed versions of AHC algorithm with different input sizes.

In Table I, we specify NewsPage.com as the text collection which is used as the source for extracting classified words, in this set of experiments. The text collection, NewsPage.com, was also used for evaluating approaches to text categorization, in previous works [4]. We extract the 300 important words from each topic for building the collection of classified words for evaluating the approaches to word clustering. We segment the entire collection which consists totally of 1200 words into the four subgroups, depending on their semantic similarities. In each category, words are selected by their frequencies concentrated on the given topic combined with subjectivity, from the text collection.

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

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

Let us mention the experimental process for validating empirically the proposed approach to the task of word clustering. We extract the important words from each category in the above text collection, and encode them into numerical and string vectors. The 1200 examples are clustered into the four clusters by the both versions of AHC algorithm. We use the clustering index which combines the two measures, the intra-cluster similarity and the inter-cluster similarity, for evaluating the both versions. The clustering index is described in detail in [24], and used previously for evaluating the clustering algorithms [1].

In Figure 8, we illustrate experimental results from clustering words using the both versions of AHC algorithm. The y-axis indicate the clustering index and is the measure for evaluating the clustering results. In the x-axis, each group indicates the input size as the dimension of numerical vectors which represent words. In each group, the gray bar and the black bar indicate the results of the traditional version and the proposed version of AHC algorithm, respectively. The most right group in Figure 8 indicates the average aver the results of the left four groups.

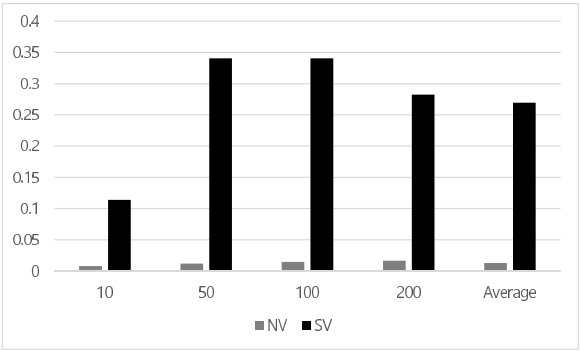

Figure 8. Results from Clustering Words in Text Collection: NewsPage.com

Let us make the discussions on the results from doing the word clustering, using the both versions of AHC algorithm, as shown in Figure 8. In the proposed version of AHC algorithm, the clustering index which is the performance measure of these clustering tasks is in the range between 0.1 and 0.34. The proposed version of the AHC Algorithm works much better in the all input sizes, as shown in Figure 8. The reason of the better performance is the improved discriminations among representations of words, by encoding words into string vectors as alternative structured forms to numerical vectors. From this set of experiments, we conclude that the proposed version works much better than the traditional one, in averaging over the four cases.

B. Opinopsis

This section is concerned with the set of experiments for validating the better performance of the proposed version: Opniopsis. In this set of experiments, the three categories are predefined in the collection, and we collect words category by category as the classified ones. A group of words is exclusively segmented into the three clusters. In this set of experiments, we also use the clustering index. Therefore, in this section, we observe the performances of the both versions of AHC algorithm with the different input sizes on another collection.

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, Opinosis, was used in previous works for evaluating approaches to

text categorization. We extract the 300 important words from each topic as the collection of classified words, for evaluating the approaches to word clustering. The group of totally 900 words is segmented into the three subgroups by the clustering algorithms, according to the number of the predefined categories. The words are extracted by both their frequencies which are concentrated in their own categories, in this set of experiments.

Table II
THE NUMBER OF TEXTS AND WORDS IN OPINIOPSIS

Category	#Texts	#Words
Car	23	300
Electronic	16	300
Hotel	12	300
Total	51	900

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 string vectors, with the input sizes: 10, 50, 100, and 200. The group of total 900 examples is clustered by the both versions of AHC algorithm into the three clusters, using the cosine similarity and the proposed one. In this set of experiments, we use also the clustering index which combines the intra-cluster similarity and the inverse inter-cluster similarity with each other, for evaluating the both versions. We adopted the external evaluation where the labeled examples are used for evaluating clustering algorithms which is mentioned in [1].

In Figure 9, we illustrate the experimental results from clustering words using the both versions of AHC algorithm. Like Figure 8, the y-axis indicates the value of clustering index, and x-axis indicates the group of the two versions of AHC algorithm by an input size. In each group, the grey bar and the black bar indicate the achievements of the traditional version and the proposed on of AHC algorithm. In Figure 9, the most right group indicates the averages over the achievements of both versions of the left four groups. Therefore, Figure 9 shows the results from clustering words into the three subgroups by both versions, on the collection: Opinosis.

We discuss the results from doing the word clustering, using the both versions of AHC algorithm, on Opinosis, shown in Figure 9. The values of clustering index of both versions range between less than 0.1 and 0.55. The proposed version of AHC algorithm works better than the traditional ones in all input sizes. The reason of its better performance is the improved discriminations among string vectors as alternative representations of words to numerical vectors. From this set of experiments, we conclude that the proposed one works outstandingly better in averaging over the four cases.

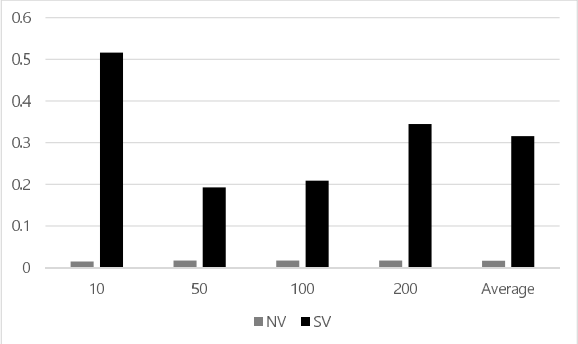

Figure 9. Results from Clustering Words in Text Collection: Opiniopsis

C. 20NewsGroups I: General Version

This section is concerned with one more set of experiments for validating empirically the better performance of the proposed version 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. The task of in this set of experiments is to cluster the gathered words into the four clusters based on their semantic similarities, exclusively. The both versions of AHC algorithm are evaluated by the clustering index, like the previous set of experiments. Therefore, in this section, we observe the performances of the both versions with the different input sizes.

In Table III, we specify the general version of 20NewsGroups which is used for evaluating the two versions of AHC 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 300 important words from them as the labeled words. In the process of gathering the classified words, they are selected by their frequencies which are concentrated in their corresponding categories. Therefore, following the external evaluation, we use the classified words for evaluating clustering results.

Table III
THE NUMBER OF TEXTS AND WORDS IN 20NEWSGROUPS I

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

The experimental process is identical is that in the previous sets of experiments. In each category, we extract the 300 important words and encode them into numerical and string vectors with the input sizes, 10, 50, 100, and 200. The totally 1200 words are clustered by the two versions

of AHC algorithm, based on their similarities. We use the clustering index which combines the intra-cluster similarity and the inverse inter-cluster similarity with each other, for evaluating the both versions, identically to the previous sets of experiments. We use the labeled words and their target labels are hidden during clustering process.

In Figure 10, we illustrate the experimental results from clustering the words using the both versions of AHC algorithm on the broad version of 20NewsGroups. Figure 10 has the identical frame of presenting the results to those of Figure 8 and 9. In each group, the gray bar and the black bar indicates the achievements of the traditional version and the proposed version of AHC algorithm, respectively. This figure presents the results from clustering words into the four clusters by changing their input sizes. We adopt the external evaluation as the paradigm of evaluating the clustering results, in this set of experiments.

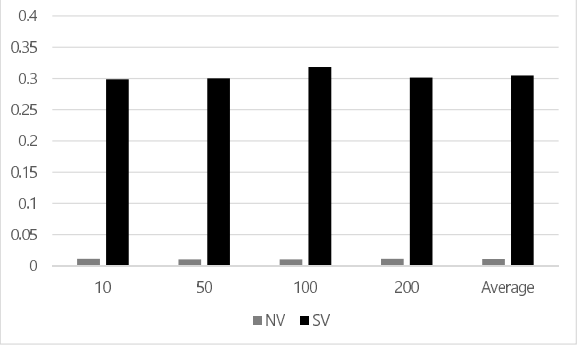

Figure 10. Results from Clustering Words in Text Collection: 20NewsGroup I

Let us discuss the results from doing the word clustering using the both versions of AHC algorithm on the broad version of 20NewsGroups, as shown in Figure 10. The clustering indices of the both versions range between less than 0.1 and 0.32. The proposed version shows the much better results in all of the input sizes. The reason of the better results is the improved discrimination among word representations. From this set of experiments, we conclude the proposed version win completely over the traditional one, in averaging their four achievements.

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 different version of 20NewsGroups. In this set of experiments, the four specific categories are predefined and words are gathered from each topic as the classified ones. The task of this set of experiments is to cluster exclusively words into four clusters. We use the clustering index like the previous sets of experiments as the evaluation metric. Therefore, in this section, we observe the performances of the both versions of AHC algorithm, with the different input sizes.

In Table 4, we specify the second version of 20NewsGroups which is used in this set of experiments. Within the general category, sci, the four categories, electro, medicine, script, and space, are predefined. We build the collection of labeled words by extracting the 300 important words from approximately 1000 texts in each specific category. In this set of experiments, the group of 1,200 words is clustered into the four groups. We use the classified words for evaluating the results from clustering them, like the case in the previous set of experiments.

Table IV
THE NUMBER OF TEXTS AND WORDS IN 20NEWSGROUPS II

Category	#Texts	#Words
Electro	1000	300
Medicine	1000	300
Script	1000	300
Space	1000	300
Total	4000	1200

The process of doing this set of experiments is same to that in the previous sets of experiments. We extract the identical number of words from all texts in each category, and encode them into numerical vectors. We cluster 1200 words by the two versions of AHC algorithm into the four clusters. We use the clustering index based on the intra-cluster similarity and inverse inter-cluster similarity, for evaluating the both versions. We evaluate the results from clustering items, using the labeled examples, following the external validity.

We present the experimental results from clustering the words using the both versions of AHC 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 11, indicates the clustering index which is used as the performance metric. In clustering words, each of them is allowed to belong to only one cluster like the cases in the previous sets of experiments.

Let us discuss the results from clustering the words using the both versions of AHC algorithm on the specific version of 20NewsGroups, as shown in Figure 11. The clustering indices of both versions range between less than 0.1 and 0.47. The proposed version shows its strongly better performances in the all input sized, as shown in Figure 11. The reason of the better performances is the discriminations among feature vectors which is improved by encoding words into string vectors, instead of numerical vectors. From this set of experiments, it is concluded that the proposed version of AHC algorithm is much feasible to the task of word clustering.

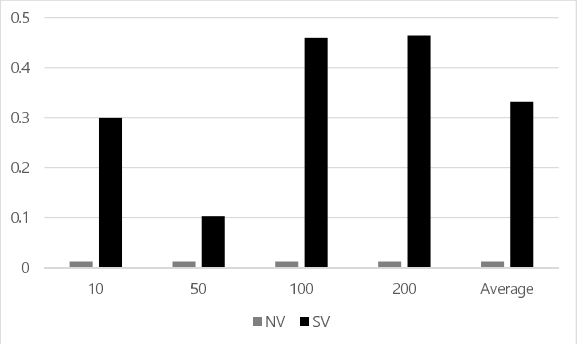

Figure 11. Results from Clustering Words in Text Collection: 20News-Group II

V. Conclusion

Let us discuss the entire results from clustering word using the two versions of AHC algorithm. In these sets of experiments, the traditional and proposed version are compared with each other in the tasks of word clustering. The proposed version shows the better results in all of the four collections. The clustering indices of the traditional version is always less than 0.1, while those of the proposed version range between 0.1 and 0.5. Through the four sets of experiments, we conclude that the proposed version improve the word clustering performance very strongly as the contribution of this research.

Let us mention the remaining tasks for doing the further research. The proposed approach should be validated and specialized in the specific domains: medicine, engineering and economics. Other features such as grammatical and posting features may be considered for encoding words into string vectors as well as text identifiers. Other machine learning algorithms as well as the AHC may be modified into their string vector based versions. By adopting the proposed version of the AHC, we may implement the word clustering system as a real program.

References

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

[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, "NTSO (Neural Text Self Organizer): A New Neural Network for Text Clustering", 31-43, Journal of Network Technology, Vol 1, No 1, 2010.

[4] T. Jo, "AHC based Clustering considering Feature Similarities", 67-70, The Proceedings of 11th International Conference on Data Mining, 2015.

[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, "Table based AHC Algorithm for Clustering Words", 574-579, The Proceedings of 18th International Conference on Advanced Communication Technology, 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, "Graph based KNN for Content based Word Classification", 24-29, The Proceedings of 12th International Conference on Multimedia Information Technology and Applications, 2016.

[10] T. Jo, "Extracting Keywords by Graph based KNN", 96-101, The Proceedings of 12th International Conference on Multimedia Information Technology and Applications, 2016.

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

[12] T. Jo, "AHC Algorithm for Text Clustering using Attribute Similarity", 148-152, The Proceedings of International Conference on Green and Human Information Technology, 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, "Keyword Extraction in News Articles using Table based K Nearest Neighbors", 1230-1233, The Proceedings of 25th International Conference on Computational Science & Computational Intelligence, 2018.

[15] 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.

[16] 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.

[17] Taeho Jo, "Improving K Nearest Neighbor into String Vector Version for Text Categorization", pp 1091-1097, ICACT Transaction on Communication Technology, Vol 7, No 1, 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, "Graph based Version for Clustering Texts in Current Affair Domain", 171-174, The Proceedings of 15st International Conference on Data Science, 2019.

[20] T. Jo, "Text Mining: Concepts and Big Data Challenge", Springer, 2019.

[21] 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.

[22] 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.

[23] T. Jo and N. Japkowicz, “Text Clustering using NTSO”, 558-563, The Proceedings of Internaitonal Joint Conference on Neural Networks, 2005.

[24] T. Jo and M. Lee, “The Evaluation Measure of Text Clustering for the Variable Number of Clusters”, 871-879, Lecture Notes in Computer Science, Vol 4492, 2007.

[25] A. Karatzoglou and I. Feinerer, “Text Clustering with String Kernels in R”, 91-98, Advances in Data Analysis, 2006.

[26] 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.

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

[28] Q. Shi, X. Qiao, and X. Guangquan, ”Using String Kernel for Document Clustering”, pp40-46, I.J. Information Technology and Computer Science, Vol 2, 2010.

[29] Y. Zheng, X. Cheng, R. Huang, and Y. Man, “A Comparative Study on Text Clustering Methods”, 644-651, Advanced Data Mining and Applications, 2006.