Similarity Metric between Tables for Modifying AHC Algorithm in Text Clustering

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 tables, instead of numerical vectors, as the approach to the text clustering. The motivations of this research are the successful results from applying the table based algorithms to the text clustering tasks in previous works and the expectation of synergy effect between the text clustering and the word clustering. In this research, we define the similarity metric between tables representing texts, and modify the AHC algorithm by adopting the proposed similarity metric as the approach to the text clustering. The proposed AHC algorithm is empirically validated as the better approach in clustering texts in news articles and opinions. In using the table based AHC algorithm, it is easier to trace results from clustering texts.**

I. Introduction

The text clustering refers to the process of segmenting a group of content based various texts into subgroups of similar ones as an instance of pattern clustering. Even if various types of approaches are available, we assume that the unsupervised machine learning algorithms are mainly used as the approaches. Texts are encoded into structured forms and clustered based on their similarities among their structured forms rather than ones among their raw texts. The text clustering results in a list of unnamed clusters and the task of naming clusters relevantly is considered as another task. Note that the clustering is a very expensive computation whatever data items are.

Let us consider the three motivations which lead to this research. First, encoding texts into numerical vector for using a traditional approach may cause the three main problems: huge dimensionality, sparse distribution, and poor transparency [4]. Second, encoding texts into tables was very successful in another task of text mining: text categorization [4]. Third, previously, we tried to encode texts into string vectors, but more mathematical definitions and characterizations were required for creating and modifying string vector based versions of machine learning algorithms [13]. Hence, the three agenda motivated us to carry out this research; we attempt to encode texts into tables for using the AHC (Agglomerative Hierarchical Clustering) algorithms. .

We present the agenda which are proposed in this research. In this research, texts are encoded into table, instead of numerical vectors, to avoid the three main problems. We define the similarity measure between tables which is always given as a normalized value and modify the AHC algorithm using the measure. The modified AHC algorithm will be used as the approach to the text clustering task. Note that each table which represents a text consists of its own entries of words and their weights.

Let us consider some benefits from this research. We avoid the three main problems in encoding texts into numerical vectors. We may expect the better performance and more stability than the traditional version of AHC algorithm. Since the table is more symbolic than the numerical vector as the representation of each text, it provides more transparency where we can guess the contents of texts only by their representations. However, since the table size is given as the external parameter of the proposed text clustering system, we need to be more careful for setting it to optimize the trade-off between the clustering reliability and the computation time.

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 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 survey previous works on the non-numerical vector based clustering algorithms. In Section II-D, we survey previous works on the clustering index which is used for evaluating clustering algorithms.

A. Related Tasks

This section is concerned with the cases of using the modernized KNN and the modernized AHC for the classification task and the clustering one. We mention the word categorization which classifies each word based on its meaning into one or some among the predefined topics, as a related task. We survey the previous cases of using the modernized

KNN algorithm for the text categorization into which the word categorization is expanded. We consider the cases of using the modernized AHC algorithm for the text clustering which is covered in this research. This section is intended to explore the previous cases of using the modernized KNN and the modernized AHC for the text clustering and its related tasks.

Let us explore the previous cases of applying the modernized KNN algorithm for the first relevant task to the text categorization. In 2016, Jo initially proposed the table based KNN version as the approach to the word categorization [16]. In 2018, he observed its better performance in comparing it with the traditional version in the word categorization [20]. The better performance of the table based version was validated in the word categorization, given as an unpublished paper [21]. In the above literatures, we mention the cases of applying the table based version of the KNN as its modernized version for the word categorization.

Let us survey the previous cases of applying the modernized KNN algorithm for the text categorization, as well as the word categorization. In 2017, it was initially asserted that the modernized KNN version which processes tables directly should be used for the text categorization [18]. In 2018, Jo started to compare the table based version with the traditional version in classifying a text in a small text collection [22]. The validation of the better performance of the table based version than the traditional one was recently finalized through the three real text collections, but it is not published, yet [33]. In the above literatures, we present the cases of applying the table based KNN version for the text categorization and the validation of its better performance.

Let us mention the previous works which are relevant directly to this research. It was initially proposed that the table based version of the AHC algorithm should be applied for clustering texts, by Jo in 2017 [19]. Its better performance of the table based version was initially observed in clustering texts in a small collection, in 2019 [30]. This research is aimed to finalize the validation of the better performance of the table based AHC algorithm in the real text collections. The clustering index which was proposed by Jo and Lee in 2007 is used as the metric for evaluating clustering results in this research [6].

We surveyed the previous cases of applying the proposed version of the AHC algorithm to the tasks which are relevant to this research. The text clustering which is covered in this research is the process of segmenting a text group into subgroups, each of which consists of content based similar texts. The AHC version which is used as the approach to the text clustering in this research is modified into the version which processes tables directly. The KNN version which was used in the word categorization and the text categorization, as well as the text clustering was modified in the same style of doing the AHC algorithm. The research about the table based AHC algorithm for clustering texts has progressed, and the goal of this research is to complete validating empirically the better performance of this version than the traditional one.

B. Encoding Schemes

This section is concerned with the previous works on various schemes of encoding texts into structured data. In this research, we propose that texts should be encoded into tables in modifying the AHC algorithm for the text clustering. We will mention the cases of encoding texts into other types of structured data: numerical vectors, string vectors, and graphs. In the literatures, which we survey in this section, we present the modifications of the AHC algorithm and the KNN algorithm into versions which process such kind of structured data, directly. This section is intended to survey the previous works on the schemes of encoding texts.

Let us survey the cases of encoding words or texts into numerical vectors, in using the modernized machine learning algorithms. In 2018, texts were encoded into numerical vectors, in using the modernized AHC algorithm for the text clustering [23]. In 2019, words were encoded into numerical vectors in using the modernized KNN algorithm for the semantic word classification [24]. In 2019, texts were encoded so in using the modernized KNN algorithm for the topic based text categorization [31]. In the above literatures, the KNN algorithm and the AHC algorithm are modernized by considering the feature similarities and the feature value similarities in computing the similarity between two numerical vectors.

Let us explore the cases of encoding texts into string vectors, each of which is an ordered finite set of strings. In 2018, the KNN algorithm was modified into the version where texts are encoded into string vectors as the approach to text categorization [25]. The modified version was applied to the text summarization which is derived from the text categorization [26]. The AHC algorithm was modified into the version where texts are encoded into string vectors, as the approach to the text clustering, in 2020 [34]. In the above literatures, we present the previous cases of encoding texts into string vectors, and modifying the machine learning algorithms, accordingly.

Let us review the previous cases of encoding words or texts into graphs for modifying the approaches to text mining tasks. In 2016, the index optimization was viewed into the task which classifies each word into expansion, inclusion, and removal, and the KNN algorithm was modified into the graph based version which processes graphs directly [17]. In 2018, the graph based version of the KNN algorithm was applied to the topic based word classification [27]. The texts were encoded into graphs for modifying the AHC algorithm into the graph based version as the approach to the text clustering [32]. In the above literatures, we presented the previous cases of encoding words or texts into graphs.

We surveyed the previous works on the structured data into which texts or words are encoded. Texts or words were encoded into numerical vectors, and the similarity metric was defined, considering the feature similarities, in order to avoid the poor discriminations among sparse numerical vectors. They were encoded into string vectors, and the semantic similarity was defined as the operation on them. They may be encoded into graphs, and the similarity between two graphs was defined. In this research, texts are encoded into tables, and the similarity metric between two tables, which is described in Section 3.2 will be defined.

C. Non-Numerical Vector based Clustering Algorithms

This section is concerned with the previous works on the non-numerical vector based clustering algorithms. The proposed version of AHC algorithm processes tables as a non-numerical vector based clustering algorithm. As more typical non-numerical vector based clustering algorithm, we will mention the table based matching clustering algorithm, the string vector based k means algorithm, and the NTSO (Neural Text Self Organizer). In the clustering algorithms which are covered in the previous works, texts are encoded into tables or string vectors, as the alternative structured data to the numerical vectors. This section is intended to explore the previous works on the three non-numerical vector based clustering algorithms.

Let us survey the previous works on the table matching clustering algorithm as the approach to the text clustering. The table matching clustering algorithm was initially proposed as the trial of preventing the problems in encoding texts into numerical vectors, in 2007 [7]. Its clustering performance was evaluated in toy experiments, using the clustering index, in 2008 [10]. The empirical validations of its performance in real experiments were finalized, in 2008 [8]. In the above literatures, we mention the table matching clustering algorithm for preventing the problems in encoding texts into numerical vectors, by encoding them into tables, as its significance.

Let us survey the previous works on the modified version of the k means algorithm which clusters string vectors, directly. It was initially modified into the string vector based version as the approach to the text clustering in 2007 [5]. Its better clustering performance than the traditional version was validated on the three text collections, in 2008 [9]. In 2010, the more desirability of encoding texts into string vectors, than doing them into numerical vectors was validated in the two clustering algorithms: the k means algorithm and the online linear clustering algorithm [11]. In the above literatures, we mention the string vector based k means algorithm as a non-numerical vector based clustering algorithm, with respect to its better performance.

Let us survey the previous works which mention the innovative neural networks, called NTSO (Neural Text Self Organizer). It was initially proposed by Jo and Japkowicz as the approach to the text clustering, in 2005 [1]. It was mentioned as an innovative neural networks, by Zheng et al. in 2006 [3]. The research on the neural networks was finalized by validating its clustering performance empirically, in 2010 [13]. The NTSO which is an unsupervised neural networks, was mentioned as a string vector based machine learning algorithm, in the above literatures.

We surveyed the previous works on the non-numerical vector based machine learning algorithms which cluster other structured data, instead of numerical vectors. We mentioned the table based matching algorithm which is used for clustering texts, and processes string vectors, directly. The string vector based k means algorithm which processes string vectors, directly, as the approach to the text clustering. The unsupervised neural networks which is called NTSO, where the input vector and the weight vectors are given as string vectors, was invented as the approach to the text clustering. The different non-numerical vector based clustering algorithm, the table based AHC algorithm, is used as the approach to the text clustering, in this research.

D. Clustering Index

This section is concerned with the previous works which deal with the clustering index as an evaluation metric. The desired directions of clustering data items are the maximization of intra-cluster similarity and the minimization of the inter-cluster similarity to clustering results. The intra-cluster similarity and the inverse inter-cluster similarity are integrated into the clustering index. The metric is used for validating empirically the table based AHC algorithm in the text clustering. This section is intended to survey of the previous works which propose and mention the clustering index.

Let us survey the previous works which propose and use the clustering index for evaluating clustering results. The clustering index was initially defined for evaluating the current quality of the document organization for implementing the DDO (Dynamic Document Organization) system, in 2006 [2]. The clustering index was proposed as the metric for evaluating the clustering algorithms, in 2007 [6]. The clustering index has been used for evaluation clustering algorithms, until now, continually [34]. In the above literatures, we mention defining and using the clustering index for evaluating the document organization and clustering algorithms.

Let us explore the previous works which cite the clustering index which is mentioned above. The clustering index was mentioned as the metric for evaluating clustering results in applying the document clustering for detecting crime patterns by Bsoul et al. in 2013 [12]. The clustering index was mentioned so in applying the ABK means algorithm for clustering documents by Gangavane et al. in 2015 [14]. It was mentioned in proposing the hierarchical co-clustering approach by Zheng et al. in 2018 [28]. In the above literatures, we present the citation of the clustering index as the metric for evaluating clustering results.

The clustering index may be used for tuning external parameters during the execution of clustering algorithms. In the above literatures, it was used for evaluating the clustering algorithms or observing the document organization. It is used for tuning the parameters of the k means algorithm and the AHC algorithm in [35]. The clustering index is mentioned as the fitness evaluation in applying the genetic algorithm for clustering data items. The advantage of tuning the parameters is the automatic optimization of parameters for maximizing the clustering quality, but its disadvantage is that it takes very much time for clustering data items.

The clustering index which was proposed and mentioned in the above previous works is adopted for comparing the proposed version of the AHC algorithm with its traditional version. The clustering index was initially used for evaluating the quality of the document organization in maintaining it. The clustering index has been used for evaluating clustering algorithms since 2007, continually. The clustering index was recently mentioned as tool for tuning external parameters of clustering algorithms during their execution. The clustering index value is always given as a normalized value between zero and one; its value which is close to one indicates the good performance.

III. Proposed Approach

This section is concerned with the AHC (Agglomerative Hierarchical Clustering) algorithm as the approach to text categorization, and it consists of the three sections. In section III-A, we describe the process of encoding a text into a table. In section III-B, we do formally that of computing a similarity between tables into a normalized value between zero and one. In section III-C, we mention the proposed version of AHC together with its traditional version. In Section III-D, we present the system architecture and the execution flow of the proposed system.

A. Text Encoding

This section is concerned with the process of encoding a text into a table. It is given as a collection of entries, and each entry consists of a word and its weight as its importance degree in the text. A table is constructed with the three steps: the text indexing, the word weighting, and the table size optimization, and each step will be explained in detail, subsequently, showing its related figures. Each table which represents a text is viewed as a set of entries, and the view is considered for computing a similarity between tables. This section is intended to describe the steps of mapping a text into a table, in detail.

The process of indexing a text into a list of words is illustrated in Figure 1. It is assumed that a single text is given as the input. The input text is transformed into a list of words by the indexing process. The basic steps of indexing a text are the tokenization, the stemming, and the stopword removal. Each step is explained in detail in [29].

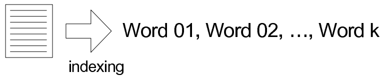


Figure 1. Text Indexing

Figure 2 illustrates computing and assigning a weight for each word in encoding a text into a table. As shown in the left side of Figure 2, a list of words is gathered from a text by indexing it. The equation which is presented in the bottom of Figure 2, is for computing the TF-IDF (Term Frequency Inverse Document Frequency) weight, and the second fields of the table are filled with the weights by computing them. Each entry of the table consists of a word and its TF-IDF weight, and the table is viewed as a set of entries. We may consider encoding a text into a table where each entry consists of multiple words and multiple weights.

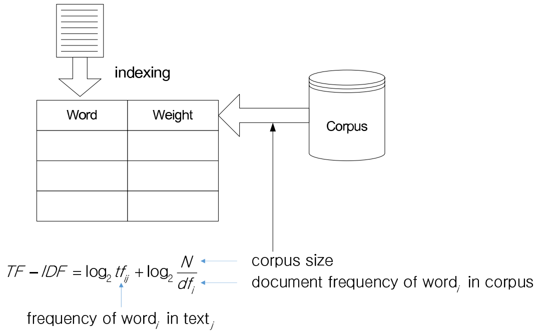

indexing
Word
Weight
Corpus
$TF-IDF = \log_2 tf_{ij} + \log_2 \frac{N}{df_i}$
corpus size
document frequency of word$_i$ in corpus
frequency of word$_i$ in text$_j$

Figure 2. Word Weighting

The process of trimming a table is illustrated in Figure 3. Because it takes the quadratic complexity to the number of entries for processing tables, it need to be downsized. The entries of the table is ranked by their weights and ones with lower weights are removed. The rank based selection and the threshold based selection become the main schemes of selecting entries. If a short text is encoded into a too small sized table, we need to consider expanding it by adding more words from external sources.

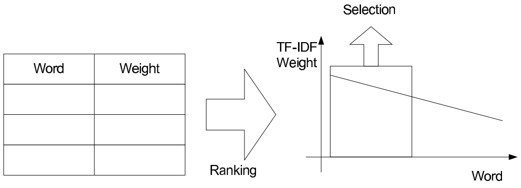


Figure 3. Table Trimming

Let us make some remarks on the process of encoding texts into tables. A table is viewed as a relational data; each record consists of a word and its weight. The frequency of a word in the text or the TF-IDF weight is used as its weight in encoding a text into a table. Because it costs the high computation complexity to their sizes for processing tables, we need to minimize the table size, keeping the reliability in performing the operations on tables. We need to define more operations on tables, for modifying other machine learning algorithms into their versions which process tables directly.

B. Similarity between Two Tables

This section is concerned with the quantified computation of the similarity between two tables. The function of a table is defined for mapping it into a set of words. The similarity between tables is computed based on entries which are shared by them. The similarity is always given as a normalized value between zero and one, and proportional to shared entries. This section is intended to describe in detail the process of computing the similarity between tables.

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

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

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

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

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

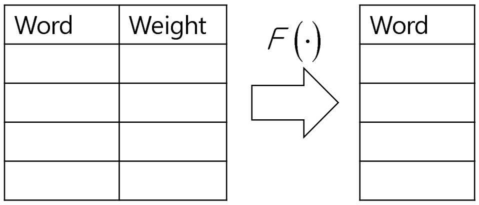


Figure 4. Mapping Table into Word Set

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

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

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

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

and the set of shared words is obtained by applying the intersection the two sets as shown in equation (3),

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

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

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

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

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

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

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

$$0 \leq sim(T_1, T_2) \leq 1 \tag{5}$$

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

Let us make some remarks on the similarity metric between tables which is described in this section. The function of a table is defined for generating a list of words which are included in it. The shared table with its entries, each of which has a shared word and its dual weights is constructed from the input tables. By equation (4), the similarity between tables is computed, and is always given as a normalized value between zero and one. The similarity metric is utilized for modifying the AHC algorithm into the table based version as the approach to the text clustering, in this research.

C. Proposed Version of AHC Algorithm

This section is concerned with the table based AHC algorithm which clusters tables, directly. In the previous section, we described the similarity metric between tables which is used for modifying the AHC algorithm, so. Texts are encoded into tables, and clustered by the AHC algorithm, depending on the similarity between clusters. We adopt the version of the AHC algorithm for implementing the text clustering system which is described in the next section. This section is intended to describe the proposed version of the AHC algorithm which clusters tables, directly.

The similarity between two tables is expanded into one between two clusters, and its computation is illustrated in Figure 4. The two clusters are notated by $C_1 = \{T_{11}, T_{12}, \ldots, T_{1|C_1|}\}$ and $C_2 = \{T_{21}, T_{22}, \ldots, T_{2|C_2|}\}$. All possible pair are generated from the two clusters, and the similarity between two tables in each pair, T_{1i} and T_{2j} by equation (4). The similarity between two clusters, C_1 and C_2, by averaging the similarities of all possible pairs, as shown in equation (7),

$$sim(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{i=1}^{|C_1|} \sum_{j=1}^{|C_2|} sim(T_{1i}, T_{2j}) \quad (6)$$

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

The process of merging two clustering into a cluster is illustrated in Figure 5. The two clusters are notated by $C_1 = \{T_{11}, T_{12}, \ldots, T_{1|C_1|}\}$ and $C_2 = \{T_{21}, T_{22}, \ldots, T_{2|C_2|}\}$ and the two clusters are assumed to be exclusive with each other, $C_1 \cap C_2 = \emptyset$. The two clusters, C_1 and C_2 are merged as expressed in equation (7),

$$merge(C_1, C_2) = \{T_{11}, \ldots, T_{1|C_1|}, T_{21}, \ldots, T_{2|C_2|}\} \quad (7)$$

If the task is a fuzzy clustering where $C_1 \cap C_2 \neq \emptyset$, The two clusters, C_1 and C_2 are merged as the union of the two clusters, as expressed in equation (8),

$$merge(C_1, C_2) = C_1 \cup C_2 \quad (8)$$

The computation of the similarity between two clusters which is mentioned above and the merge of two clusters are the main operations in proceeding the data clustering by the AHC algorithm.

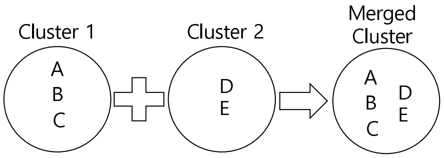
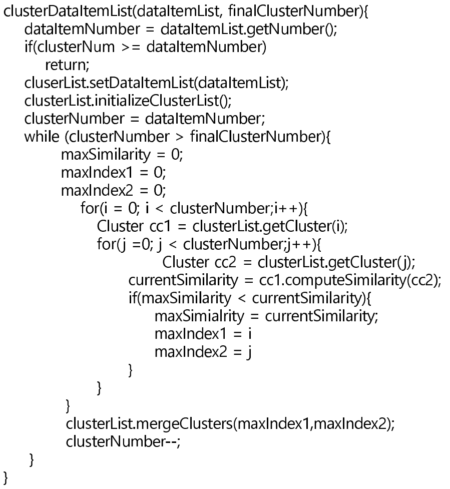


Figure 5. Merge of Two Clusters

In Figure 6, the process of clustering data items by the AHC algorithm is illustrated. This algorithm is initialized with singletons as many as data items. The similarities of all possible pairs which are generated from clusters are computed, and the cluster pair with the highest similarity is merged into a cluster. The two steps are iterated until the desired number of clusters which is given as the external parameter of this algorithm. The number of clusters is decreased for every iteration by one.

Let us make some remarks on the proposed version of the AHC algorithm which clusters table directly. Texts are encoded into tables by the process which was described in Section III-A, and the similarity metric which was described in Section III-B, is used for computing the similarity between clusters. The similarities of all possible pairs of clusters are computed, and clusters in the pair with its highest similarity are merged into a cluster. The AHC algorithm is executed by iterating the similarity computation and the cluster merge until the desired number of clusters. The difference of the proposed AHC algorithm from the traditional version is to use the similarity metric between two tables, instead of the cosine similarity or the Euclidean distance which are ones between numerical vectors.



```
clusterDataItemList(dataItemList, finalClusterNumber){
    dataItemNumber = dataItemList.getNumber();
    if(clusterNum >= dataItemNumber)
        return;
    cluserList.setDataItemList(dataItemList);
    clusterList.initializeClusterList();
    clusterNumber = dataItemNumber;
    while (clusterNumber > finalClusterNumber){
            maxSimilarity = 0;
            maxIndex1 = 0;
            maxIndex2 = 0;
               for(i = 0; i < clusterNumber;i++){
                  Cluster cc1 = clusterList.getCluster(i);
                  for(j =0; j < clusterNumber;j++){
                             Cluster cc2 = clusterList.getCluster(j);
                        currentSimilarity = cc1.computeSimilarity(cc2);
                        if(maxSimilarity < currentSimilarity){
                            maxSimialrity = currentSimilarity;
                            maxIndex1 = i
                            maxIndex2 = j
                        }
                  }
             }
             clusterList.mergeClusters(maxIndex1,maxIndex2);
             clusterNumber--;
     }
}
```

Figure 6. Process of Clustering Data Items by AHC Algorithm

D. Text Clustering System

This section is concerned with the system architecture and the execution flow of the text clustering system in its design level. The AHC algorithm which was described in Section III-C is adopted for implementing the text clustering system. Texts which are given as clustering targets are encoded into tables by the process of which is described in Section III-A, and they are clustered by the AHC algorithm. In this section, we present the system architecture and the execution flow, but omit the implementation in Java or Python. This section is intended to describe the proposed system in its design level.

The group of texts and tables is illustrated in Figure 7. All of the texts are initially given at a time under the assumption of the clustering as the offline one. Texts in the group are encoded into tables by the process which was described in Section III-A. They are clustered by the AHC algorithm which was described in Section III-C. The online clustering where texts are given as a stream will be considered in the next research.

The system architecture of the text clustering system is illustrated in Figure 8. The encoding module encodes the texts which are given as the clustering targets into tables by the process which was described in Section III-A. The clustering module which has the similarity computation module as its nested ne for computing the similarity between clusters, clusters graphs by the AHC algorithm which was

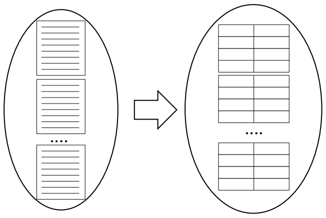


Figure 7. Encoding Texts into Tables

described in Section III-C. The decoding module restores texts from the graphs. The M clusters of texts which are presented in Figure 8 become the final output of the system.

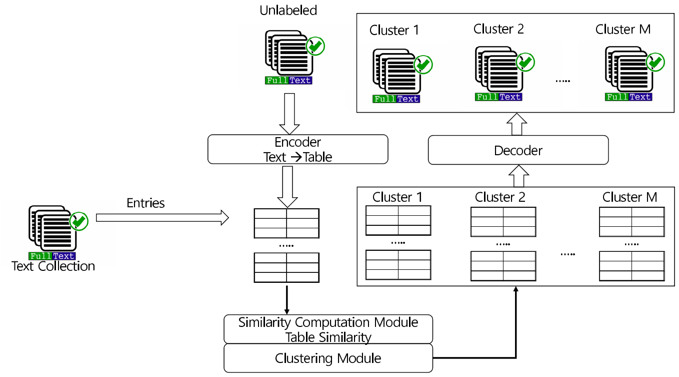


Figure 8. System Architecture

The execution flow of the text clustering system is illustrated in Figure 9. Texts which are given as clustering targets are encoded into tables. The similarity between tables which was described in Section III-B is used for computing the similarity between clusters. Data items are clustered by iterating the computation of cluster similarities and the merge of clusters. Text clusters are generated as the final output of the system.

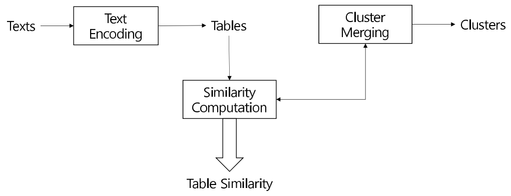


Figure 9. System Architecture

Let us make some remarks on the system architecture and the execution flow of the text clustering system which are presented in Figure 8 and 9. Encoding texts into tables and the similarity between tables are proposed in this research. The AHC algorithm is modified by defining the similarity metric which is described in Section III-B as one between clusters, and the modified AHC algorithm is applied for implementing the text clustering system. In this research, the system architecture and the execution flow which are needed for doing the general design of the system are presented in this research. The detail design and the implementation of the system will be considered in the next research.

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 text clustering on the collection, NewsPage.com. In Section IV-B, we show the results from applying it for clustering texts 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 texts from 20NewsGroups.

A. *NewsPage.com*

This section is concerned with for validating empirically 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 clustering results, and gather texts from the collection, category by category as labeled ones. Each text is allowed to be arranged into only one of the four clusters, in proceeding the clustering task, in this set of experiments. We use the clustering index which was proposed in [2] for evaluating the clustering results. Therefore, this section is intended to observe the performance of both versions of AHC algorithm with the different input sizes.

In Table I, we specify the text collection, NewsPage.com, which is used in this set of experiments. The text collection was used for evaluating approaches to text categorization in previous works [15]. In this collection, the four categories are predefined: Business, Health, Internet, and Sports, and we select 300 texts at random in each category. The entire group which consist of 1200 texts is segmented into four subgroups by clustering algorithm, in this set of experiments. This collection was built by copying and pasting news articles from the web site, newspage.com, in 2005, as plain text files.

Table I
THE NUMBER OF TEXTS IN NEWSPAGE.COM

Category	#Texts	#Used Texts
Business	500	75
Health	500	75
Internet	500	75
Sports	500	75
Total	2000	300

Let us mention the experimental process for validating empirically the proposed approach to the task of text clustering. In each category, we select the 300 texts among totally the 500 texts, and encode them into numerical vectors and tables. The group of 1200 texts is segmented into the four clusters by the two versions of AHC algorithm. We use the clustering index which combines the intra-cluster similarity and inter-cluster similarity, for evaluating the both versions of AHC algorithm. The detail description of the clustering index is provided in [6], and it was previously used for evaluating the clustering results.

In Figure 1, we illustrate the experimental results from clustering texts, using the both versions of AHC algorithm. The y-axis indicate the clustering index as the measure for evaluating the clustering results. In the x-axis, each group indicates the input size which is the dimension of numerical vectors which represent texts. In each group, the gray bar and the black bar indicates the achievements of the traditional version and the proposed version of AHC algorithm, respectively. The two bars in the most right group indicates the averages over their results of the four left groups.

Let us make the discussions on the results from doing the text clustering, using the both versions of AHC algorithm, as shown in Figure 1. The clustering index which is the performance measure of these clustering tasks is in the range between and 0.1 and 0.5. The proposed version of AHC algorithm works strongly better in all input sizes as shown in Figure 1. The reason of the better results of the proposed version is the improve discriminations among representations by encoding texts into tables, instead of numerical vectors.

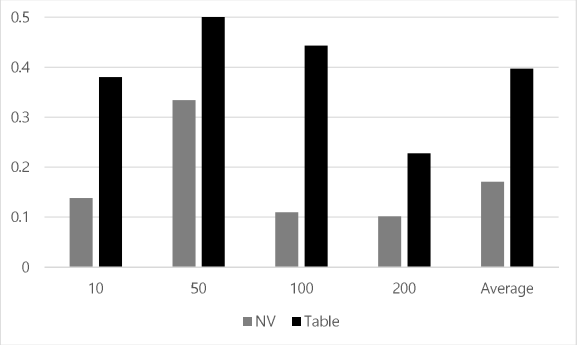


Figure 10. Results from Clustering Texts in Text Collection: NewsPage.com

From this set of experiments, we conclude that the proposed version works better than the traditional one, in averaging over the four cases.

B. Opinopsis

This section is concerned with the set of experiments for validating the better performance of the proposed version on the collection, Opinosis. We set the number of clusters as three, following the number of the predefined categories, and prepare the labeled texts from the collection. The entire group of collected texts is exclusively clustered into the three subgroups. We use the clustering index as the evaluation measure. In this section, we observe the performances of the both versions of AHC algorithm with the different input sizes.

In Table II, we specify the text collection, Opinosis, which is used in this set of experiments. The text collection was used in previous works, for evaluating the approaches to text categorization. The three categories, ‘Car’, ‘Electronics’, and ‘Hotel’, are predefined and all texts are used for evaluating the approaches to text clustering, in this set of experiments. The group of total 51 texts is exclusively segmented into the three clusters as many as the predefined categories. We obtained the collection by downloading it from the web site, http://archive.ics.uci.edu/ml/machine-learning-databases/opinion/.

Table II
THE NUMBER OF TEXTS IN OPINIOPSIS

Category	#Texts	#Used Texts
Car	23	23
Electronic	16	16
Hotel	12	12
Total	51	51

We perform this set of experiments by the process which is described in section IV-A. We use all of 51 texts which are labeled with one of the three categories and encode them into numerical vectors and tables with the input sizes: 10, 50, 100, and 200. The group of total 51 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 [2].

In Figure 11, we illustrate the experimental results from clustering texts using the both versions of AHC algorithm. Like Figure 10, the y-axis indicates the value of clustering index, and the x-axis indicates the group of two versions by an input size. In each group, the gray bar and the black bar indicate the results of the traditional version and the proposed version, respectively. In Figure 11, the most right group indicates the averages over the results of the left four groups. Therefore, the Figure 11 shows the results from clustering texts into the three subgroups by the both versions of AHC algorithm, on the text collection, Opinosis.

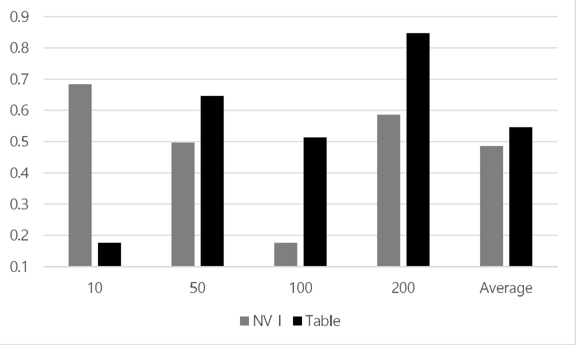


Figure 11. Results from Clustering Texts in Text Collection: Opiniopsis

We discuss the results from doing the text clustering using the both versions of AHC algorithm on Opinosis, shown in Figure 11. The values of clustering index of both versions range between 0.1 and 0.9. The proposed version works better than the traditional one in the two input sizes: 50, 100, and 200. The clustering index of the proposed version reaches even more than 0.9 in the input size, 200. From this set of experiments, we conclude the proposed version works outstandingly better than the traditional version in averaging the four cases.

C. 20NewsGroups I: General Version

This section is concerned with one more set of experiments for validating the better performance of the proposed version on the text collection, 20NewsGroups I. In this set of experiments, we predefine the four general categories in this collection, and gather texts from it in each predefined one as the classified ones. The task of this set of experiments is to cluster texts into the four subgroups based on their semantic similarities, exclusively. We evaluate the both versions of AHC algorithm by clustering index. Therefore, in this section, we observe the performances of the both versions with the different input sizes.

This section is concerned with one more set of experiments for validating the better performance of the proposed version on the text collection, 20NewsGroups I. In this set of experiments, we predefine the four general categories in this collection, and gather texts from it in each predefined one as the classified ones. The task of this set of experiments is to cluster texts into the four subgroups based on their semantic similarities, exclusively. We evaluate the both versions of AHC algorithm by clustering index. 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 300 texts from 4000 or 5000 texts at random. Following the external evaluation, we use the classified words for evaluating clustering results. We obtain the collection, 20NewsGroup, by downloading from the web site, https://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html, as one of the standard text collection for evaluating approaches to text categorization.

Table III
THE NUMBER OF TEXTS IN 20NEWSGROUPS I

Category	#Texts	#Used Texts
Comp	5000	300
Rec	4000	300
Sci	4000	300
Talk	4000	300
Total	17000	1200

The experimental process is identical is that in the previous sets of experiments. In each category, we extract the 300 texts at random and encode them into numerical vectors and tables with the input sizes, 10, 50, 100, and 200. The totally 1200 texts are clustered into the four subgroups 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 texts and their target labels are hidden during clustering process.

In Figure 12, we illustrate the experimental results from clustering the texts into the four groups on the broad version of 20NewsGroups. Figure 12 has the identical frame of presenting the results to those of Figure 10 and 11. 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. Figure 12 presents the results from clustering texts by changing their input sizes. In this set of experiments, we adopt the external evaluation as the paradigm of evaluating the clustering results.

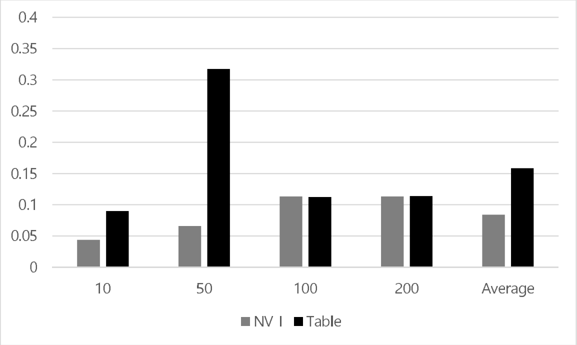


Figure 12. Results from Clustering Texts in Text Collection: 20NewsGroups I

Let us discuss the results from doing the text clustering using the both versions of AHC algorithm, as shown in Figure 12. The clustering indices of the both versions range between 0.05 and 0.32. The proposed version shows its outstandingly better performance in the two input sizes: 10 and 50. It keeps its competitive performance in the others. From this set of experiments, we conclude that the proposed version wins over the traditional version, in averaging the 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 version of 20NewsGroups. In this set of experiments, the four specific categories are predefined in this collection. Texts are exclusively clustered into the four subgroups like the previous sets of experiments. We use the clustering index as the metric for evaluating clustering results. Therefore, in this section, we observe the performances of the both versions of AHC algorithm with the different input sizes.

In Table IV, we specify the specific version of 20NewsGroups which is used as the test collection, in this set of experiments. Within the general category, sci, we predefine the four categories: ‘electro’, ‘medicine’, ‘script’, and ‘space’. In each category, we select 300 texts among approximately 1000 texts, at random. We evaluate the results from clustering texts by the clustering index which is used as the evaluation metric, in the previous sets of experiments. We use the classified texts for evaluating the results, hiding their labels, while clustering texts.

The process of doing this set of experiments is same to that in the previous sets of experiments. We select the balanced number of texts from the collection over categories, and encode them into the representations with the input sizes which are identical to those in the previous set of experiments. Using the two versions of AHC algorithm, we cluster the 300 examples into the four clusters, identically to the previous set of experiments. We use the clustering index whose bases are the intra-cluster similarity and the inverse inter-cluster similarity, for evaluating the both versions of AHC algorithm. We evaluate the results from clustering items, using the labeled examples, following the external validity.

Table IV
THE NUMBER OF TEXTS IN 20NEWSGROUPS II

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

We present the experimental results from clustering the texts using the both versions of KNN algorithm on the specific version of 20NewsGroups. The frame of illustrating the classification results is identical to the previous ones. In each group, the gray bar and the black bar stand for the achievements of the traditional version and the proposed version, respectively. The y-axis in Figure 13, indicates the clustering index which is used as the performance metric. In clustering texts, each of them is allowed to belong to only one cluster like the cases in the previous sets of experiments.

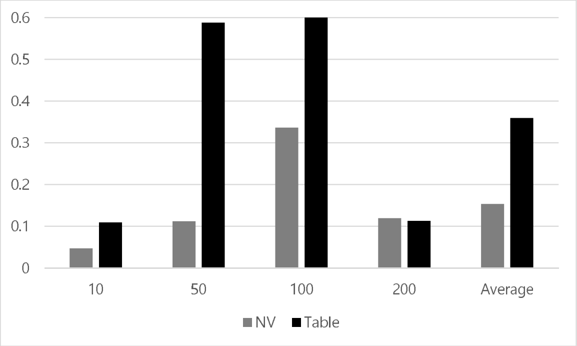


Figure 13. Results from Clustering Texts in Text Collection: 20NewsGroups II

Let us discuss on the results from clustering the texts on the specific version of 20NewsGroups, as shown in Figure 13. The accuracies of the both versions range between 0.05 and 0.6. The proposed version shows its outstandingly better performance in three of the four input sizes. It keeps its comparable one in the input size, 200. From this set of experiments, it is concluded that the proposed version shows its better performance by averaging over the accuracies of the four cases.

V. Conclusion

Let us discuss the entire results from performing text clustering using the two versions of KNN algorithm. The both versions is compared with each other in the task of text clustering, in these sets of experiments. The proposed version show its better results in the four collections. The clustering indices of the traditional version range between 0.09 and 0.69, while those of the proposed version range between 0.12 and 0.82. From the four sets of experiments, we conclude that the proposed version improves the text clustering performance, as the contribution of this research.

We need the remaining tasks for doing the further research. We may apply the proposed approach for clustering texts in the specific domains such as medicine, law, and engineering. We may consider the semantic relations among different words in the tables in compute their similarities, but it requires the similarity matrix or the word net for doing so. We may install the process of optimizing weights of words as the meta-learning tasks. We may implement the text clustering system, adopting the proposed approach.

References

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

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

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

[4] T. Jo and D. Cho, “Index Based Approach for Text Categorization”, 127-132, International Journal of Mathematics and Computers in Simulation, No 2, 2007.

[5] T. Jo, M. Lee, and T. M. Gatton, “Inverted Index based Operation on String Vector for K Means Algorithm in Text Clustering”, 267-271, The Proceedings of International Conference on Machine Learning: Models, Technologies & Applications, 2007.

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

[7] K. Yi, T. Jo, M. Lee, and Y. Choi, “Modifying Online Text Clustering Algorithm using Inverted Index based Operation”, 150-153, The 2007 International Conference on Semantic Web and Web Services, 2007.

[8] T. Jo, “Single Pass Algorithm for Text Clustering by Encoding Documents into Tables”, 1749-1757, Journal of Korea Multimedia Society, Vol 11, No 12, 2008.

[9] T. Jo, “Inverted Index based Modified Version of K-Means Algorithm for Text Clustering”, 67-76, Journal of Information Processing Systems, Vol 4, No 2, 2008.

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

[11] T. Jo, “Modification of Clustering Algorithms for Text Clustering”, 21-33, International Journal of Computer Science and Software Technology, Vol 3, No 1, 2010.

[12] Q. Bsoul, J. Salim, and L.Q. Zakaria, “An intelligent document clustering approach to detect crime patterns”, 1181-1187, Procedia Technology, 2013.

[13] 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.

[14] H. N. Gangavane, M. C. Nikose, P. C. Chavan, “A novel approach for document clustering to criminal identification by using ABK-means algorithm”, 1-6, The Proceedings of IEEE International Conference on Computer, Communication and Control, 2015.

[15] T. Jo, “Normalized Table Matching Algorithm as Approach to Text Categorization, pp839-849, Soft Computing, Vol 19, No 4, 2015.

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

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

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

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

[20] 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.

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

[22] 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.

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

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

[25] T. Jo, “Improving K Nearest Neighbor into String Vector Version for Text Categorization”, 1091-1097, ICACT Transaction on Communication Technology, Vol 7, No 1, 2018.

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

[27] T. Jo, “Comparing Graph based K Nearest Neighbor with Traditional Version in Word Categorization in NewsPage.com”, 12-18, International Journal of Advanced Social Sciences, Vol 1, No 1, 2018.

[28] L. Zheng, Y. Qu, X Qian, and G. Cheng, “A hierarchical coclustering approach for entity exploration over Linked Data”, 200-210, Knowledge Base Systems, Vol 142, 2018.

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

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

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

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

[33] T. Jo, “Table based K Nearest Neighbor for Text Classification”, unpublished, 2020.

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

[35] T. Jo, “Machine Learning”, Springer, 2021 (Scheduled).