Content based Word Clustering using Feature Similarity based AHC Algorithm

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

***Abstract*—This article proposes the modified AHC (Agglomerative Hierarchical Clustering) algorithm which considers the feature similarity and is applied to the word clustering. The texts which are given as features for encoding words into numerical vectors are semantic related entities, rather than independent ones, and the synergy effect between the word clustering and the text clustering is expected by combining both of them with each other. In this research, we define the similarity metric between numerical vectors considering the feature 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. The significance of this research is to improve the clustering performance by utilizing the feature similarities.**

I. Introduction

The word clustering refers to the process of segmenting a group of words into subgroups of content based similar words. The group of words is encoded into their structured forms and a similarity measure between them is defined. The words are arranged into their closet clusters based on the similarity measure, as the clustering proceeds. The results from clustering the words are unnamed clusters and cluster naming and cluster prototype definition are regarded as other tasks in this research. The scope of this research is restricted to cluster words by their meanings.

Let us mention some challenges which this research tries to solve. The strong dependency among features exists especially in the text mining tasks, so the Bayesian networks which considers it was proposed as the approach, but it requires very much complicated analysis for using it [**?**]. If the independences among features are assumed, it requires many features for encoding words or texts into numerical vectors. Since each feature has very little coverage in the domain of text mining, we cannot avoid the sparse distribution of numerical vectors which represent words or texts[**?**]. Therefore, this research is intended to solve the problems by considering the feature similarity as well as the feature value one.

Let us mention what we propose in this research as its idea. In this research, we consider the both similarity measures, feature similarity and feature value similarity, for computing the similarity between numerical vectors. The AHC (Agglomerate Hierarchical Clustering) algorithm is modified into the version which accommodates the both similarity measures. The modified version was applied to the word clustering task. Therefore, the goal of this research is to improve the word clustering performance by solving the above problems.

Let us mention the benefits which we expect from this research. The consideration of both the feature similarity and the feature value similarity provides the way of reducing the dimensionality of numerical vectors, potentially. We discover semantic relations among words through this research for performing other text mining tasks. The improvement of discriminations among even sparse numerical vectors is caused by computing the similarity between numerical vectors using the two measures. Therefore, the goal of this research is to pursue the benefits for implementing the text clustering systems.

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 semantic operations which are relevant to the process of computing the feature similarity. In Section II-C, we survey the previous works on schemes of computing the similarity between texts which corresponds to the feature similarity. In Section II-D, we survey the previous works on the clustering index which is used for evaluating the approaches to the word clustering.

A. Related Tasks

This section is concerned with the previous cases of using the modernized KNN and the modernized AHC for the tasks which are relevant to the word clustering. We will mention the word categorization which classifies words based on their meanings as a related task. Because the word clustering

is the task which is covered in this research, we mention the cases of using the modernized AHC algorithms to the word clustering. We consider the keyword extraction which is derived from the word categorization as a related task where the modernized KNN algorithms are applied. This section is intended to survey the previous cases of applying the modernized AHC and KNN to the word clustering and its related tasks.

Let us survey on the previous cases of applying the KNN algorithm which is modernized by considering both the feature similarities and the feature value ones to the word categorization as the first relevant task. In 2015, Jo initially proposed the idea of modifying the KNN algorithm by considering the feature similarities [7]. In 2018, he tried to compare the modernized KNN algorithm with the traditional one [21]. In 2018, he validated completely its better performance in the three test sets: NewsPage.com, Opniopsis, and 20NewsGroups [22]. In the above literatures, we present the effectiveness of the modernized KNN algorithm in the word categorization.

Let us explore the cases of applying the modernized AHC algorithm for the word clustering which is covered in this research. In 2015, Jo initially proposed it by describing the idea of the modified AHC algorithm as the approach to the word clustering [8]. In 2018, its better results of the proposed AHC algorithm were observed in a very small collection of texts as a toy experiment, in clustering words [23]. In 2007, Jo and Lee proposed the metric which is used for evaluating clustering results [2]. This research is intended to finalize the empirical validation of the better results of the modernized AHC algorithm in the semantic word clustering.

Let us mention the cases of using the modernized KNN algorithm for the keyword extraction which is derived from the word categorization. The idea of implementing the keyword extraction system using the modernized KNN algorithm was initiated by Jo in 2015 [9]. The better results of the modernized version was initially discovered in comparing it with the traditional version in the keyword extraction in a small text collection, by Jo in 2018 [11]. The empirical validation of the better performance than the traditional version was finalized, but it not published, yet [26]. In the above literatures, we explored the previous cases of using the modernized KNN algorithm for the keyword extraction.

We surveyed the cases of applying the modernized KNN algorithm and the modernized AHC algorithm to the tasks which are relevant to this research. The word clustering which is covered in this research is the task where words are clustered based on their meanings. The modernized AHC algorithm which is adopted as the approach to the word clustering is one where a similarity between words is computed, considering the feature similarities. The modernized KNN algorithm is used as the supervised learning algorithm in the word categorization and the keyword extraction which are mentioned above. The proposed version of the AHC algorithm will be validated empirically, using the evaluation metric which was proposed in 2007, in clustering words in the real text collections.

B. Semantic Operations

This section is concerned with the previous works on the semantic operations on strings. It is defined as the operation on strings under the assumption of each string with its own meaning. The semantic similarity which is the base operation on the strings is for generating a semantic similarity between two strings as a normalized value between zero and one. The semantic similarity is expanded into the semantic similarity mean for analyzing strings statistically and semantically. This section is intended to survey the previous works on the three semantic operations.

Let us explore the previous works where the semantic similarity is defined, and applied for modifying the machine learning algorithms. The semantic similarity between two strings was defined and expanded into the string vector kernel function, in modifying the SVM (Support Vector Machine), in 2008 [3]. The semantic similarity was applied for implementing the NTSO (Neural Text Self Organizer) which is the string vector based unsupervised neural networks, in 2010 [4]. It was applied for modifying the KNN algorithm into its string vector based version in 2018 [24]. In the above literatures, we present that the semantic similarity was defined as the base semantic operation on strings, and used for modifying the machine learning algorithms.

Let us mention the semantic operation, SSM (Semantic Similarity Mean) in [10]. It is the semantic operation which is derived from the semantic similarity and is for averaging semantic similarities of all possible pairs of strings. In this operation, any number of strings, each of which has its own meaning, is given as the input, and the mean similarity over strings between zero and one is given as the output. The output value from this operation indicates the cohesion of the word group; highly averaged similarity is given to the group of semantically similar words. This operation is used for tuning clustering algorithm parameters in the word clustering.

Let us mention the semantic operation which generates a single set, instead of a numerical value. We mentioned above the two semantic operations, the semantic similarity and the SSM, which generate a normalized value between zero and one. The semantic operation which generates a single set was defined for creating the neural networks which are called NTSO (Neural Text Self Organizer), in 2010 [4], and in the operation, strings which have more similarities between two strings which are given as the operands are retrieved as a set. The operation was used for updating the weights which are given as strings between two layers of the neural networks. In [4], the neural networks, NTSO, processes string vectors, each of which consists of strings, instead of numerical values, directly.

We surveyed the previous works on the semantic operations on strings. The semantic similarity is used for computing the feature similarity in this research. The operation, SSM, on strings is derived from the semantic similarity for analyzing semantically a group of strings. The semantic string set which generates a set of strings which are more similar as the two input strings was defined for creating the unsupervised neural networks, called NTSO. The semantic similarities of the d features are computed using the semantic similarity, and the similarity matrix where its rows and columns correspond to the d features, is constructed.

C. Text Similarities

This section is concerned with the previous works which deals with the similarities between texts. The features are given as texts in encoding words into numerical vectors. Because the similarity between texts is computed as a feature similarity in this research, we need to survey the previous works on the similarity metric between texts. The similarity between texts was computed in the previous works which are surveyed in this section, by encoding texts into structured data such as tables, string vectors, and graphs. This section is intended to explore the previous works about the schemes of computing the similarity between texts.

Let us survey the previous works where the similarity between texts is computed by encoding them into tables. The similarity between tables which represent texts was used for modifying the KNN algorithm as the approach to the text categorization, in 2016 [12]. The similarity between tables was computed in applying the KNN algorithm to the text summarization, in 2016 [13]. The AHC algorithm was modified as the approach to the text clustering by the similarity metric between tables, in 2016 [14]. In the above literatures, we present that the similarity between texts is computed by encoding them into tables.

Let us explore the previous works on the computation of the similarity between texts by encoding them into string vectors. The KNN algorithm was modified using the scheme of computing the similarity between texts into the version where texts are encoded into strings as the approach to the text categorization in 2016 [15]. The scheme of computing the similarity between texts was used for modifying the AHC algorithm as the approach to the text clustering in 2016 [16]. The KNN version was applied to the text summarization in [17]. In the above literatures, we present the computation of the similarity between texts by encoding them into string vectors.

Let us survey the previous works on the scheme of computing the similarity between texts by encoding them into graphs. The similarity between graphs was defined as the similarity between texts, and the KNN algorithm was modified into its graph based version, using it, as the approach to the text categorization in 2016 [18]. The graph based version of the KNN algorithm was applied to the text summarization which is mapped into a classification task, in 2016 [19]. The AHC algorithm was modified into the graph based version, using the similarity between graphs, as the approach to the text clustering, in 2016 [20]. In the above literatures, we presented the similarity between graphs which is used as one between texts.

We surveyed the previous works on the schemes of computing the similarity between texts. Texts are given as features in encoding words into numerical vectors, so in this research, we need the scheme of computing the similarity between texts as the feature similarity. In the above literatures which are surveyed in this section, the similarity between texts is computed by encoding texts into tables, string vectors, or graphs. The similarity metrics between tables, between string vectors, and between graphs, were defined as ones between texts, and they were used for modifying the KNN algorithm and the AHC algorithm. In this research, the similarity between texts is computed by indexing them into word sets, based on the intersection of two word sets.

D. Clustering Index

This section is concerned with the previous works on the clustering index which is the metric for evaluating clustering results. The desired direction of clustering data items is to maximize the intra-cluster similarity which is the cohesion of each cluster and to minimize the inter-cluster similarity for maximizing the discrimination among clusters. The clustering index is the evaluation metric which integrates the both similarity, following the style of doing the precision and the recall into the F1 measure. The clustering index may be used for tuning the external parameters of clustering algorithms, as well as for evaluating the clustering results. This section is intended to explore the previous works on the clustering index which is used for evaluating clustering algorithms, in this research.

Let us survey the cases of proposing and using the clustering index as the metric for evaluating the clustering results. The clustering index was initially proposed by Jo in 2006, as the mean of evaluating quantitatively the quality of the document organization [1]. The clustering index was used as the metric for evaluating clustering algorithms by Jo and Lee in 2007 [2]. It has been used for evaluating clustering algorithm until recent year, continually [27]. In this research, it is used as the metric of evaluating clustering algorithms.

Let us survey the previous works which cites the clustering index as the metric or evaluating clustering results. In 2013, Bsoul et al. mentioned the clustering index as one of main evaluation metrics in applying the document clustering for detecting the crime patterns [5]. It was mentioned in proposing the ABK means algorithm as the approach to the document clustering, by Gangavane et al. in 2015 [6]. The clustering index was used for evaluating hierarchical co-

clustering results by Zheng et al. in 2018 [25]. The clustering index which was proposed by Jo was cited as the evaluation metric to clustering results, in the above literatures.

Let us consider using the clustering index for tuning external parameters of clustering algorithms. The current clustering results are evaluated with the clustering index based on the intra-cluster similarity and the inter-cluster similarity. The module of tuning the parameters with the clustering index was installed to the k means algorithm and the AHC algorithm [28]. The genetic algorithm may be applied for clustering data items by using the clustering index as the fitness value [28]. The external parameters are optimized automatically, but it takes much more time for clustering data items as the payment.

In this research, we adopt the clustering index which was mentioned in the previous works for evaluating the clustering algorithms. The clustering index was considered for maintaining the document organization, dynamically. It was utilized for evaluating clustering algorithms, continually. It was used for tuning external parameters of clustering algorithms during their executions. The clustering index value is always given as a normalized value between zero and one.

III. Proposed Approach

This section is concerned with modifying the AHC (Agglomerative Hierarchical Clustering) algorithm into the version which considers the similarities among features as well as feature values, and it consists of the three sections. In Section III-A, we describe the process of encoding words into numerical vectors. In Section III-B, we do formally the proposed scheme of computing the similarity between two numerical vectors. In Section III-C, we mention the proposed version of AHC algorithm which considers the similarity among features as the approach to word clustering. In Section III-D, we present the architecture and the execution flow of the proposed system.

A. Word Encoding

This section is concerned with the process of encoding a word into a numerical vector. A list of words are given as the input, and the texts which include one among them are taken as the feature candidates. Some are selected among them as the features by the text length or the total word frequency. For each word, the values are assigned to the features, for building a numerical vector as its representation. This section is intended to describe the process of encoding a word into a numerical vector, step by step.

The process of generating feature candidates which are given as texts from a corpus in Figure 1. In the initial stage, the corpus as a text collection and K words as encoding targets are prepared. For each word, texts which include itself, are retrieved from the corpus, and it is linked to a set of texts. The union is performed on the text sets which are linked to words; the texts in the union set become the feature candidates. More than ten thousands feature candidates are generated under the assumption that a corpus contains more than 20,000 texts.

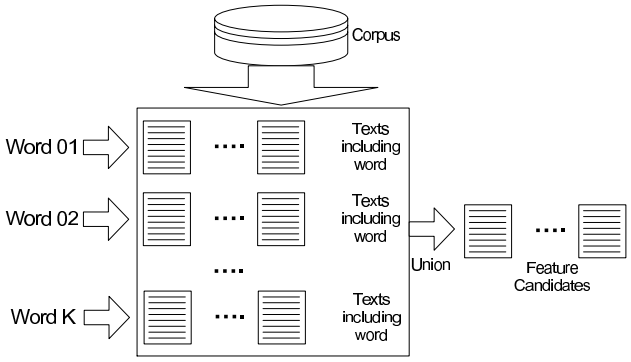

Figure 1. The Process of generating Feature Candidates

Figure 2 shows the selection of d texts as the features among N texts where d ¡¡ N. The N texts are gathered in the previous process which is illustrated in Figure 1. The selection criteria is defined and only d texts are selected as features based on the criteria. The text size, the total frequency of K words, or the total weight of them usually become the selection criteria. The number of feature candidates, N, is ten thousands, and the number of selected features, d, is hundreds, in general.

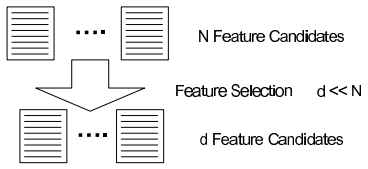

Figure 2. The Process of selecting Features

The schemes of assigning values to the selected features in encoding a word into a numerical value are illustrated in Figure 3. A binary value, zero or one, which indicate whether the word is included in the text, or not, may be assigned to each feature. A relevancy frequency as the rate of frequency of the word in the text to the total frequency, may be assigned to each feature. A TF-IDF (Term Frequency and Inverse Document Frequency) weight which is computed by the equation which is presented in Figure 3, may be assigned to each features. The feature value may be adjusted by the posting properties and the grammatical functions of the words in the text.

Let us make some remarks on the process of encoding words into numerical vectors. Texts in a corpus are used as features for mapping words into numerical vectors. A value which is assigned to a feature indicates a relationship between a text corresponding to a feature and a word as an encoding target. Because each feature has very weak

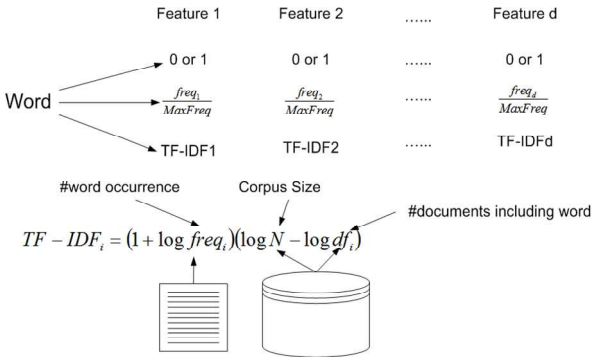

Figure 3. The Process of weighting Word

coverage, numerical vectors which represents words tend to be sparse. The dimension of numerical vectors which represent words or texts is usually three hundreds.

B. Similarity Metric

This section is concerned with the proposed similarity metric between two vectors. The cosine similarity and the inverse Euclidean distance are typical traditional similarity metric between two vectors. The cosine similarity is modified by introducing the feature similarity which is one among features of numerical vectors. The features are given as texts in encoding a word into a numerical vector, and the similarity between two texts are given as a feature similarity for computing the similarity between two numerical vectors. This section is intended to describe the proposed similarity metric, which considers the feature similarity.

The frame of computing the similarity between two numerical vectors is illustrated in Figure 4. In the traditional similarity metric, the only feature value similarities as the similarity between elements of two numerical vector with the one to one matching are considered. In the proposed similarity metric, additionally, the feature similarities which are ones among the features of numerical vectors, $f_1, f_2, \ldots, f_d$ with all possible pair matchings are considered. In order to avoid the poor discriminations among sparse numerical vectors, the similarity between numerical vectors is computed, considering both the feature similarities and the feature value similarities. As the payment of the proposed computation scheme, the linear complexity is increased to the quadratic complexity.

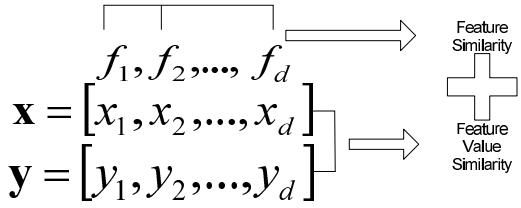

Figure 4. Frame of computing Similarity between two Numerical Vectors

The similarity matrix of the features which are given as texts is illustrated in Figure 5. The d texts, $text_1, text_2, \ldots, text_d$are selected as the features by the process which was described in Section III-A. A text is indexed into a list of words; T_i and T_j are sets of words respective from $text_i$ and $text_j$. The similarity between two texts is computed by equation (1),

$$sim(text_i, text_j) = \frac{2|T_i \cap T_j|}{|T_i| + |T_j|} \quad (1)$$

The similarity between two texts is given as a feature similarity based on the rate of the shared words to ones in either of texts, as shown in equation (1).

$$\begin{array}{c} \\ text_1 \\ text_2 \\ \cdots \\ text_d \end{array} \begin{array}{c} \begin{array}{cccc} text_1 & text_2 & \cdots & text_d \end{array} \\ \begin{bmatrix} s_{11} & s_{12} & \ldots & s_{1d} \\ s_{21} & s_{22} & \ldots & s_{2d} \\ \ldots & \ldots & \ldots & \ldots \\ s_{d1} & s_{d2} & \ldots & s_{dd} \end{bmatrix} \end{array} \quad s_{ij} = sim(text_i, text_j)$$

Figure 5. Similarity Matrix

Let us derive the equation for computing the proposed similarity metric with the feature similarities. Equation (1) is notated into its simplified form as expressed in equation (2),

$$sim(text_i, text_j) = f_{ij} \quad (2)$$

The two words are encoded into the two d dimensional numerical vectors, $\mathbf{x} = [x_1\ x_2\ \ldots\ x_d]$ and $\mathbf{y} = [y_1\ y_2\ \ldots\ y_d]$.

The similarity between the two numerical vectors is computed by equation (3),

$$sim(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^{d} \sum_{j=1}^{d} f_{ij} \cdot x_i \cdot y_j}{d\|\mathbf{x}\|\|\mathbf{y}\|} \quad (3)$$

where $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^{d} x_i^2}$ and $\|\mathbf{y}\| = \sqrt{\sum_{i=1}^{d} y_i^2}$. It takes the quadratic complexity to the d dimensional vector for computing the similarity by equation (3), as the payment.

Let us make some remarks on the similarity metric which is proposed in this research for modifying the KNN algorithm. The feature similarity which is the similarity between features and the feature value similarity which is the similarity between the values in the numerical vectors are considered as the frame of computing the similarity between numerical vectors. The d texts are given as the features, the similarities of all possible pairs from them are computed, and the similarity matrix which consists of the feature similarities as its elements is constructed. As the payment, it takes the quadratic complexity, $O(d^2)$, for avoiding the poor discriminations among sparse numerical vectors, as the payment.

C. Proposed Version of AHC Algorithm

This section is concerned with the proposed version of the AHC algorithm. The proposed similarity metric between numerical vectors was described in the previous section, and used for modifying the AHC algorithm. The idea of the proposed AHC algorithm is to compute the similarity between two clusters, using the similarity which was described in the previous section, in proceeding the clustering. The benefit from the proposed AHC algorithm is to improve the clustering performance by solving the poor discriminations among sparse numerical vectors. This section is intended to describe the proposed version of the AHC algorithm.

The computation of the similarity between clusters is illustrated in Figure 6. Each cluster consists of numerical vectors, and the mean vectors are computed in the two clusters. The similarity between two mean vectors of clusters is computed by equation (3), as the similarity between two clusters. Here, we know that the proposed similarity metric which is expressed in equation (3) is use for modifying the AHC algorithm. The similarity between clusters is always given as a normalized value like one between two numerical vectors.

The process of merging two clustering into a cluster is illustrated in Figure 7. The two clusters are notated by $C_1 = \{\mathbf{x}_{11}, \mathbf{x}_{12}, \ldots, \mathbf{x}_{1|C_1|}\}$ and $C_2 = \{\mathbf{x}_{21}, \mathbf{x}_{22}, \ldots, \mathbf{x}_{2|C_2|}\}$ 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 (4),

$$merge(C_1, C_2) = \{\mathbf{x}_{11}, \ldots, \mathbf{x}_{1|C_1|}, \mathbf{x}_{21}, \ldots, \mathbf{x}_{2|C_2|}\} \quad (4)$$

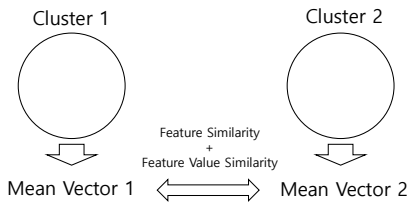

Figure 6. Similarity between Clusters

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 (5),

$$merge(C_1, C_2) = C_1 \cup C_2 \quad (5)$$

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.

The process of clustering data items by the AHC algorithm is illustrated in Figure 8. The algorithm is initialized by singletons as many as data items. The similarities of all possible pairs of clusters are computed, and the pair of clusters with its highest similarity is merged into a cluster. The two steps are repeated until the desired number of clusters. The number of clusters is decreased by one in every iteration, and the desired number of clusters is given as an external parameter of this algorithm.

Let us make some remarks on the proposed version of the AHC algorithm, as the approach to the word clustering. The cosine similarity is replaced by the similarity metric which was described in Section III-B, for computing the

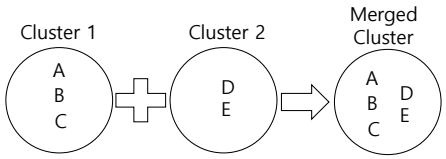

Figure 7. Merge of Two Clusters

```
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 8. Process of Clustering Data Items by AHC Algorithm

similarity between clusters. Two clusters with the highest similarity is are computed into a cluster; elements in the both clusters belong to one cluster. The AHC algorithm is executed by iterating the computation of the similarities of all possible pairs of clusters and merge of the pair with its highest similarity into one. The desirable number of clusters is reach by decreasing the number of clusters by one.

D. Word Clustering System

This section is concerned with the system architecture of the word clustering system. The AHC algorithm which uses the similarity metric which is described in Section III-B, is adopted and was described in detail in Section III-C. The words which are given as the clustering targets are encoded into numerical vectors, and clustered into subgroups, each of which consists of semantically similar ones, by the AHC algorithm. The system architecture and execution flow of the system are presented; the detail implementation of the proposed system in the source code will be considered in the next research. This section is intended to describe the system architecture and the execution flow of the word clustering system.

The process of encoding words which are given as clustering targets into numerical vectors is illustrated in Figure 9. In implementing the word clustering system, it is assumed that words are encoded into numerical vectors, by the process which was described in Section III-A. In the system, the numerical vectors which represent words are clustered by the AHC algorithm which was described in Section III-C. The online clustering where words are given as a continual stream will be considered in the next research.

The system architecture of the word clustering system is illustrated in Figure 10. In the encoding module, words are encoded into numerical vectors. They are clustered by the AHC algorithm which was described in Section III-C, and the similarity computation module is included for computing the similarity between numerical vectors by the process which was described in Section III-B, in the clustering module as the core part of the system. From the module, the clusters of numerical vectors are generated, and they are decoded into words. From the system, the word clusters are generated as the final output.

The execution flow of the word clustering system is illustrated in Figure 11. Words are encoded into numerical

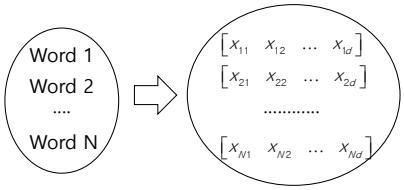

Figure 9. Word Encoding

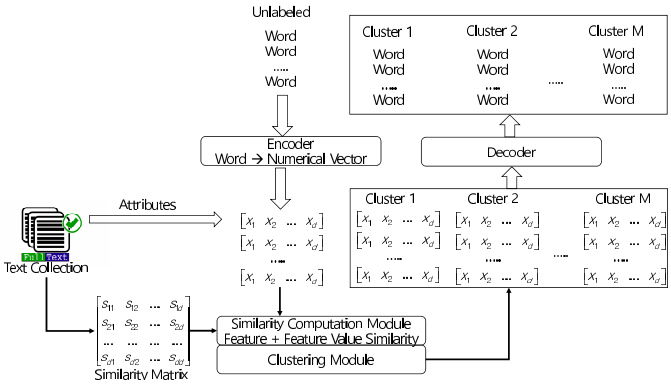

Figure 10. System Architecture

vectors, and the similarity matrix is constructed from the text collection for computing the feature similarities. The similarity metric which was described in Section III-B was used for computing the similarity between clusters. Data items are clustered by iterating computing cluster similarities and merging clusters with their highest similarity. When reaching the desired number of clusters, the execution of the system is terminated.

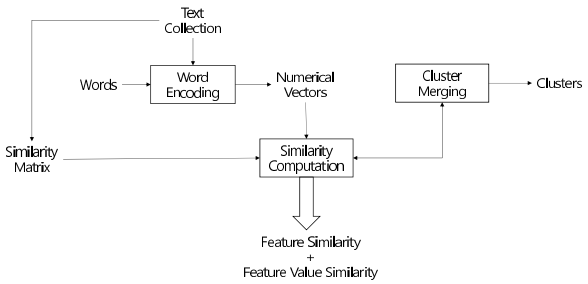

Figure 11. Execution Process

Let us make some remarks on the architecture and the execution flow of the word clustering system. We proposed the similarity metric between numerical vectors which is tolerant to the sparse distribution over them, and adopted it for computing the similarity between clusters in the AHC algorithm. It is proposed that the modified AHC algorithm which was described in Section III-C was applied for implementing the word clustering system. This research provides the system architecture and the execution process which are necessary for make the general design of the system. In the next research, we will cover the detail design and the implementation in Java or Python.

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 [**?**] 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 [**?**]. 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 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 [2], and used previously for evaluating the clustering algorithms [**?**].

In Figure 12, 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 12 indicates the average aver the results of the left four groups.

Let us make the discussions on the results from doing the word clustering, using the both versions of AHC algorithm, as shown in Figure 12. 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 0.35. The proposed version of the AHC Algorithm works much better in the all input sizes, as shown in Figure 12. The reason of the better performance is the improved discriminations among feature vectors representing words, by considering the feature similarities as well as feature value ones. From this set of experiments, we conclude that the proposed version works much better than the traditional one, in averaging over the four cases.

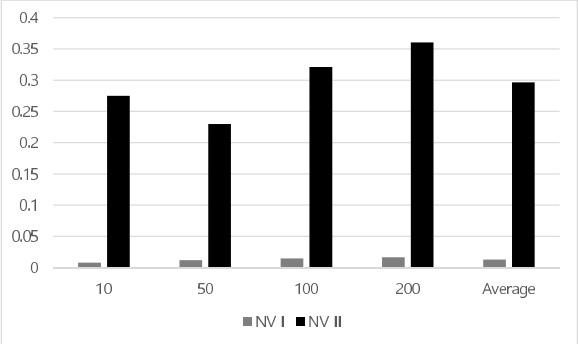

Figure 12. Results from Clustering Words in Text Collection: NewsPage.com

B. Opinopsis

This section is concerned with the set of experiments for validating the better performance of the proposed version: 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, with 10, 50, 100, and 200 dimensions. 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 [**?**].

In Figure 13, we illustrate the experimental results from clustering words using the both versions of AHC algorithm. Like Figure 12, 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 13, the most right group indicates the averages over the achievements of both versions of the left four groups. Therefore, Figure 13 shows the results from clustering words into the three subgroups by both versions, on the collection: Opinosis.

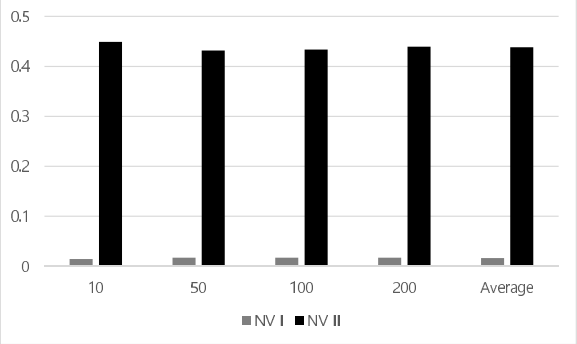

Figure 13. Results from Clustering Words in Text Collection: Opiniopsis

We discuss the results from doing the word clustering, using the both versions of AHC algorithm, on Opinosis, shown in Figure 13. The values of clustering index of both versions range between less than 0.1 and 0.5. The proposed version of AHC algorithm works better than the traditional ones in all input sizes. The reason of its better performance is the discriminations among feature vectors represent words which are improved by considering feature similarities as well as feature value ones. From this set of experiments, we conclude that the proposed one works outstandingly better in averaging over the four cases.

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 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 14, we illustrate the experimental results from clustering the words using the both versions of AHC algorithm on the broad version of 20NewsGroups. Figure 14 has the identical frame of presenting the results to those of Figure 12 and 13. 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.

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 14. The clustering indices of the both versions range between less than 0.1 and 0.5. 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

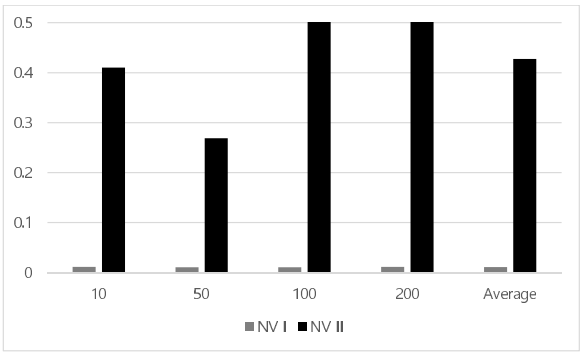

Figure 14. Results from Clustering Words in Text Collection: 20News-Group I

representations by considering the feature similarities. 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 20News-Groups 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 15, 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..

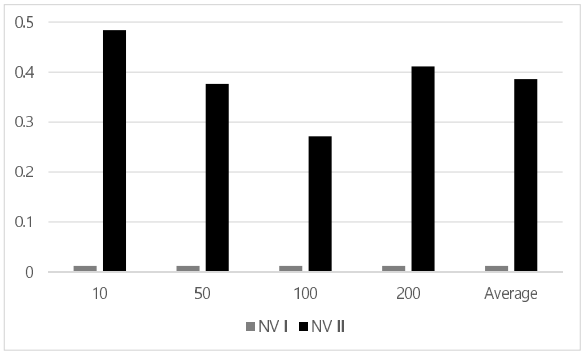

Figure 15. Results from Clustering Words in Text Collection: 20News-Group II

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 15. The clustering indices of both versions range between less than 0.1 and 0.12. The proposed version shows its strongly better performances in the all input sized, as shown in Figure 4. The reason of the better performances is the discriminations among feature vectors which is improved by considering the feature similarities as well as feature value ones. From this set of experiments, it is concluded that the proposed version of AHC algorithm is much feasible to the task of word clustering.

V. CONLUSION

Let us mention the remaining tasks for doing the further research. We need to validate the proposed approach in specific domains such as medicine, engineering, and economics, as well as in generic domains such as ones of news articles. We may consider the computation of similarities among some main features rather than among all features for reducing the computation time. We try to modify other machine learning algorithms such as Naive

Bayes, Perceptrons, and SVM (Support Vector Machine) based on both kinds of similarities. By adopting the proposed approach, 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] Taeho Jo and Malrey Lee, "The Evaluation Measure of Text Clustering for the Variable Number of Clusters", pp871-879, Lecture Notes in Computer Science, Vol 4492, 2007.

[3] T. Jo, "Modified Version of SVM for Text Categorization", 52-60, International Journal of Fuzzy Logic and Intelligent Systems, Vol 8, No1, 2008.

[4] 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.

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

[6] 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.

[7] T. Jo, "KNN based Word Categorization considering Feature Similarities", 343-346, The Proceedings of 17th International Conference on Artificial Intelligence, 2015.

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

[9] T. Jo, "Keyword Extraction by KNN considering Feature Similarities", 64-68, The Proceedings of The 2nd International Conference on Advances in Big Data Analysis, 2015.

[10] 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.

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

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

[13] T. Jo, "Table based AHC Algorithm for Clustering Words", 574-579, The Proceedings of 18th International Conference on Advanced Communication Technology, 2016.

[14] T. Jo, "Table based KNN for Extracting Keywords", 812-817, The Proceedings of 18th International Conference on Advanced Communication Technology, 2016.

[15] T. Jo, "Encoding Words into String Vectors for Word Categorization", 271-276, The Proceedings of 18th International Conference on Artificial Intelligence, 2016.

[16] T. Jo, "String Vector based AHC as Approach to Word Clustering", 133-138, The Proceedings of 12th International Conference on Data Mining, 2016.

[17] T. Jo, "Using String Vector based KNN for Keyword Extraction", 27-32, The Proceedings of 15th International Conference on Advances in Information and Knowledge Engineering, 2016.

[18] 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.

[19] 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.

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

[21] T. Jo, "Word Classification in Domain on Current Affairs by Feature Similarity based K Nearest Neighbor", 348-351, The Proceedings of International Conference on Artificial Intelligence, 2018.

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

[23] Taeho Jo, "Modification of AHC algorithm for Clustering Words into Feature Similarity based Version", pp359-362, The Proceedings of International Conference on Artificial Intelligence, 2018.

[24] 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.

[25] L. Zheng, Y. Qu, X Qian, and G. Cheng, "A hierarchical co-clustering approach for entity exploration over Linked Data", Knowledge Base Systems, 200-210, Vol 142, 2018.

[26] T. Jo, "Extracting Keywords from Text by Feature Similarity based K Nearest Neighbor", unpublished, 2020.

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

[28] T. Jo, "Machine Learning", Springer, 2021 (Scheduled).