Graph and Centroid-based Word Clustering

Santipong Thaiprayoon
FernUniversität in Hagen
Hagen, Germany
santipong.thaiprayoon@fernuni-hagen.de

Herwig Unger
FernUniversität in Hagen
Hagen, Germany
herwig.unger@gmail.com

Mario Kubek
FernUniversität in Hagen
Hagen, Germany
mario.kubek@fernuni-hagen.de

ABSTRACT

With the recent exponential growth of text documents, a word clustering algorithm is an essential approach for making a reduction in a huge amount of text data and unsupervised feature selection on the domain of natural language processing. This paper proposes a novel method of the graph and centroid-based word clustering. The proposed method aims to automatically group similar words into the same cluster and handles noisy text and outliers. The proposed method applies the concepts of the hierarchical agglomerative clustering and K-means algorithm to find similar words according to the criterion of distance range on the co-occurrence graph. The small clusters and isolated words are also merged into another cluster. The experimental results demonstrate that the proposed method consistently and significantly outperforms state-of-the-art baselines in word clustering algorithms on the ground truth dataset. Besides, the proposed method is unsupervised learning and generic, which could be applied to various tasks of natural language processing and text mining.

CCS Concepts

• Computing methodologies→ Artificial intelligence → Natural language processing→ Natural language generation.

Keywords

Word clustering; Word co-occurrence; Natural language processing; Graph-based model; Hierarchical agglomerative clustering;

1. INTRODUCTION

Due to the explosive growth of a significant number of textual data from publicly available sources such as discussion forums, books, publications, and web pages over the last decades [1], tools and mechanisms for word clustering algorithms are an essential task for reducing the amount of textual data and obtaining useful information. This task is the practical benefit for solving real-word applications in the domain of natural language processing (NLP) such as thesaurus construction, information retrieval, text classification, statistical language modeling, and word sense disambiguation.

Word clustering is a process of automatically partitioning a set of words into a set of classes called clusters (groups or categories) [2]. The most significant part of the word clustering algorithm is choosing the words that have distinction and similar meaning. Each cluster contains the most similar words based on their word similarity in structure of contextual, syntactic, or semantic and dissimilar to words in distinct groups. With the basic assumption that the words within a cluster are similar to other words in the same cluster and dissimilar to the words in other clusters. In other words, intra-cluster distances are minimized, and inter-cluster distances are maximized. Moreover, the word clustering algorithm has become a useful technique for unsupervised feature selection and dimension reduction, especially when dealing with the sparse data problem. The main goal of this technique is to enhance the accuracy and performance in various domains of natural language processing and text mining.

Several research studies have been conducted and provided their practical approaches to address the construction of word clusters, which are based on different techniques [3], [4], [5]. Most word clustering algorithms used the n-gram model and greedy word clustering statistical technique to create a set of clusters of words, which do not take semantics into account. In the classical clustering algorithms still require the number of clusters for the initialization of centroids. In the other direction, the word clustering algorithm is widely used on a variety of real-world applications such as content-based recommendation systems, text categorization, text summarization, information retrieval, and text classification.

Even though these research papers have achieved promising performance in several applications, they have a major drawback of specifying a certain number of clusters to be the initialization of centroids. Most traditional word clustering is sensitive to the initialization of centroids, which often leads to a lack of the inconsistency of the final result of clusters. For minor limitations, previous methods cannot address sensitive to noisy text and outliers, which result in the ineffective quality of clusters. For the others, the position or sequence of words in the document is also ignored and cannot preserve the semantic relationship of words.

To tackle this challenge, this paper proposes a novel method of unsupervised word clustering using a graph model and a centroid-based approach. The proposed method consists of two main phases. In the first phase, the main idea is to apply the hierarchical agglomerative method and K-means clustering to group similar words according to the criteria of distance range on the graph. The graph-based model assists the proposed method in capturing the implicit structure of the textual data. The reason for combining both two approaches with their advantages is to improve efficiency and the clustering quality. The hierarchical agglomerative clustering is a technique that does not require the number of clusters and can address sensitive to noisy data and outliers. For calculating the distance between words in a graph, Dijkstra's algorithm, the most common solution in graph theory, is utilized to find the shortest path between two words. The distance from the source to a destination is calculated as a cost of traversing between a pair of words in a graph. The total of the

Conference'10, Month 1–2, 2010, City, State, Country.

DOI: http://dx.doi.org/10.1145/12345.67890

lowest cost is then used for determining important words of a text corpus. For the second phase, the small clusters and isolated word clusters are then iteratively merged according to a value of distance range until convergence criteria or no words moved to groups. Besides, this method also finds the number of word clusters automatically without selecting the number of clusters. All in all, the outputs of the proposed method are a collection of word clusters.

The main contribution of this research article consists of four aspects. Firstly, an unsupervised word clustering algorithm based on graph theory is proposed. The proposed method combines the concepts of both hierarchical agglomerative clustering and K-means algorithm, a conceptual representation of the co-occurrence graph to enhance the quality of word clusters. One of the advantages of the proposed method is based on an undirected weighted co-occurrence graph. A center cluster in the graph is represented as a centroid term. The centroid term is determined with the minimum average distance to all words of the documents in the graph. This method calculates the relationship between words based on words co-occurrence statistics. Moreover, this method also does not specify a certain number of clusters to be generated. Secondly, this method plays an important role in the task of natural language processing such as text classification, information retrieval, and text summarization. The key idea is to use a set of clusters of words as unsupervised feature representations that can reduce dimensions, computational complexity, and improve efficiency and accuracy. Thirdly, the proposed method is an unsupervised approach that does not require the training data, and language and domain-independent entities. Finally, this study investigates the hypothesis that the proposed method is significantly superior to other existing algorithms in word clustering that can find meaningful word clusters efficiently. The proposed method also solves the limitations presented in state-of-the-art algorithms. The method, moreover, is capable of applying effectively in various tasks of natural language processing and text mining.

The remaining part of the research article is organized as follows. Section 2 presents a literature survey that describes the related previous works. Section 3 discusses the proposed model in great detail. Section 4 describes the process of experiments. Results with discussion are presented in Sections 5, and Section 6 concludes the work and indicates some future research directions.

2. LITERATURE REVIEWS

In this section, several major topics are described as fundamental concepts for developing word clustering algorithms, including a brief overview of the well-known algorithms of clustering. Next, the basic definitions of finding centroid terms on the co-occurrence graph are explained. Then, previous research works are discussed in an aspect of the limitations of word clustering algorithms.

2.1 K-means

K-Means algorithm [6] is one of the most popular unsupervised learning algorithms that solve well-known clustering problems. This algorithm is a hard clustering that partitions several data points into a cluster by iteratively updating the cluster centers and the associated data points. Therefore, the data points within the same cluster are similar to other data points within the same cluster than those in other clusters. Each data point is assigned to a cluster based on a minimum distance between a data point and a centroid point. Distance measures are employed to calculate similarity and dissimilarity between the data points. Each cluster has a cluster center or centroid that is most representative of the cluster. It assigns data point to a cluster that the sum of squared distance between the data point and the cluster center is minimal from all the data points of a particular cluster.

2.2 K-means++

One problem of the standard K-means algorithm is sensitive to the initialization of centroids. This sensitivity leads to limitations in the accuracy of the final result of clusters. Another limitation is that the initialization of centroids is critical to the quality of determining the optimal number of clusters. To overcome the drawbacks, K-means++ [7] is designed to improve the centroid initialization for the classical K-means algorithm. The basic assumption is that initial centroids should be distant from each other. This assumption increases the chances of partitioning entirely different clusters. K-means++ aims to use the furthest point strategy that leads to significant improvement of the quality of clusters and time reduction to convergence.

2.3 DBSCAN

Density-based spatial clustering of applications with noise (DBSCAN) [8] is a data clustering algorithm, which density-based clustering non-parametric algorithm, given a set of data points in search space. Density-based has the feature to group data points that are closely packed together (points with many nearby neighbors), marking as outlier points that lie alone in low-density regions (whose nearest neighbors are too far away).

2.4 LSI

Latent semantic analysis (LSA) [9] is a technique in natural language processing, particularly distributional semantics, of analyzing relationships between a set of documents and the terms. They contain by producing a set of concepts related to the documents and terms. LSA assumes that words that are close in meaning will occur in similar pieces of text (the distributional hypothesis). A matrix containing word counts per document (rows representing unique words and columns representing each document) is constructed from a large piece of text. A mathematical technique called singular value decomposition (SVD) is used to reduce the number of rows while preserving the similarity structure among columns.

2.5 Building a Co-occurrence Graph

The co-occurrence graph is a basic model to obtain information about text documents than term frequency vectors [10]. The frequency of occurrence of two terms in a sentence from a document is called co-occurrence. The idea is that words that occur together in the same context will be close in meaning. This means that similar words will occur together in a piece of text. In the next step, each of the sentences is tokenized into a set of words. To define similarity among the sentences, a co-occurrence graph G = (V, E) is built, where V is a set of vertices represented as unique words. E is a set of edges representing the relationship between a pair of words. A weight of the connection between two words is added to the corresponding edge in graph G. The edge is weighted by the frequency of word co-occurrence. A weight on edge indicates the significance of the association between words in a document.

2.6 Finding Centroid Terms

A centroid is a crucial term that could represent text as the center of a document. To identify centroid terms of a text document, a minimum average distance is used by calculating the distance to all words in the cluster from the cluster center [11].

2.7 Survey of Research Works

In recent years, several studies for the word clustering algorithms have been reported in the literature. The objective of the algorithm is to automatically partition a set of words into clusters from heterogeneous sources such as email messages, discussion forums, books, publications, and web pages. Yutaka Matsuo et al. [12] proposed a graph-based word clustering algorithm using the web search engine. The pointwise mutual information and chi-square are used to measure as a similarity for the clustering algorithm and showed that chi-square measure outperforms the result of the pointwise mutual information of the words. They also evaluated the clustering on two sets of words derived from the web directory and WordNet. King-Ip Lin et al. [13] proposed word-based soft clustering (WBSC), an efficient soft clustering algorithm based on a given similarity measure. WBSC used a hierarchical approach to cluster documents having similar words. Jiguang Liang et al. [2] presented an unsupervised algorithm for word clustering based on a probabilistic transition matrix. A text document dataset, a list of words is generated by removing stop words and unimportant words. Each word is required to be represented by the documents in the dataset, which results in a co-occurrence matrix. For calculating the similarity of words, a word similarity graph with transition probabilities as weight edges is created. Then, a new kind word clustering algorithm, based on label propagation, is applied. Yuan Lichi [14] presented a novel concept of word similarity by using mutual information based on word similarity, the concept of the word set similarity was given, and a bottom-up hierarchical clustering algorithm. Nikoletta Bassiou et al. [15] developed two techniques for word clustering algorithms, which employed long-distance bigram language models. The first technique is built on a hierarchical clustering algorithm and minimizes the sum of Mahalanobis distances of all words after a cluster combination from the centroid of the class created by merging. The second technique applied the probabilistic latent semantic analysis. Both techniques could create more compact word clusters when either long-distance bigrams or their interpolated versions are employed rather than when the classical bigrams are used.

From previous research, most of the existing word clustering algorithms use the vector space model, which performs documents as bags of words. As a result, the position or sequence of words in the documents is ignored. Meanwhile, the bag-of-words technique cannot preserve the semantic meaning of words that have different meanings depending on the part of speech. This paper differs from the aforementioned research works that the proposed method generates a set of clusters of words on the co-occurrence graph. The method applies the concepts of the hierarchical agglomerative algorithm and K-means clustering to find similar words according to the criterion of the distance range on the graph. The key objective is to improve the accuracy and performance of word clustering algorithms. The proposed method is thus the main tool for several tasks of natural language processing, which is useful for unsupervised feature selection, dimension reduction, and decrease in the amount of textual data.

3. PROPOSED METHODOLOGY

This section describes methods for building a set of word clusters. The process overview of the proposed method is explained and illustrated in Figure 1.

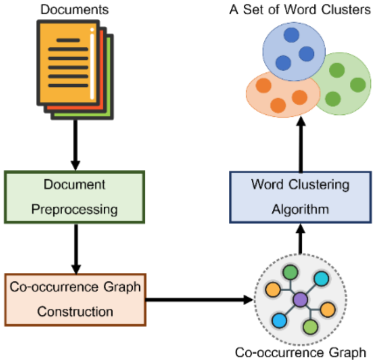

Figure 1. The process overview of the proposed method

The key objective of the proposed method is to propose a novel word clustering algorithm based on a combination of the hierarchical agglomerative method and K-means clustering with a conceptual representation of the undirected weighted co-occurrence graph to enhance the quality of the generated word clusters. The results of the proposed method are a collection of word clusters that contains the most similar words which share a similar concept or semantic meanings. The proposed method consists of three components: (1) Document preprocessing, (2) Co-occurrence graph construction, and (3) The word clustering algorithm, where each step is given in detail below.

3.1 Document Preprocessing

To prepare the input document for the following tasks, the proposed method includes several preprocessing steps presented in the detail as follows.

- *Splitting the text document into sentences*: In this step, the text document is split into a set of sentences based on a neural network with detecting terminators from the StanfordNLP package. After finishing this step, each document is represented as a set of sentences denoted by $D = \{S_1, S_2, . . ., S_n\}$.
- *Tokenizing sentences*: Each sentence $S \in D$ is treated as a set of tokens, denoted by $S = \{w_1, w_2, . . ., w_k\}$. Besides, each token is turned to lowercase to facilitate the subsequent processing tasks.
- *Normalizing word*: In this step, stemming and lemmatization are applied to process individual words in a sentence by reducing different forms of words to their base form.
- *Removing stop words and unimportant tokens*: A standard list of stop words is created, and these stop words are used to filter out stop words in each sentence. Stop words mean the most common words that have less meaning, including punctuation marks, spaces, and word terminators.
- *Extracting noun*: The part-of-speech tagging (POS) is used to label each sentence, which contains a list of parts of speech. This study aims to identify and select the only noun in the sentence.

After completing this component, a set of sentences is sent to the next component to create a co-occurrence graph.

3.2 Co-occurrence Graph Construction

To construct the undirected weighted co-occurrence graph, Figure 2 illustrates the process of building the graph, which can be explained in the details below.

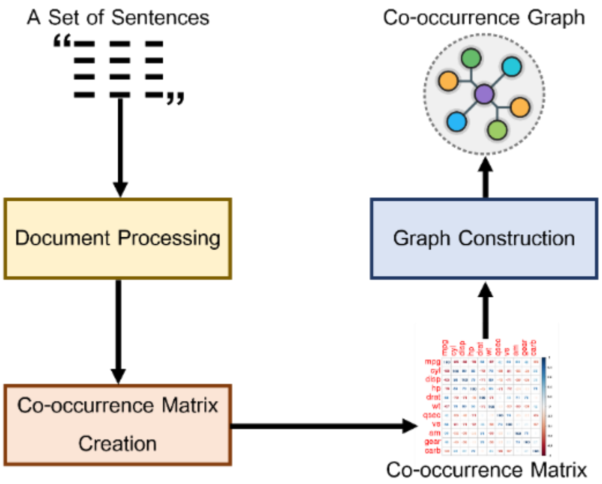

Figure 2. The co-occurrence graph construction

The process of this component is started using a set of sentences derived from the previous component. The co-occurrence matrix is then created as text representation. Words co-occurrence statistics on the matrix is computed from the frequency of occurrence of two terms in sentences. The technique generates semantic and syntactic relationships. After that, the co-occurrence graph is built by inserting words, edges, and weights into the graph.

The documents are represented as an undirected weighted graph G = (V, E, W). Graph models are a way of representing information by encoding it in vertices and edges. The nodes V is a set of nodes representing a word that occurs in the documents. The edges E is a set of edges between every pair of words. To create a graph representation, the most common is an adjacency matrix for the textual graph representing the weights of edges. The weighted W is a set of weights assigning to the edges of the graph G. The edges are weighted by a distance score that represents the strength of the relationship of the connected words. The distance score calculates the correlation coefficient between words based on co-occurrence statistics of words. The distance score is word co-occurrence that appears together in a sentence. The calculation of the distance score $d(w_i, w_j)$ between two words can be defined by equation 1.

$$d(w_i, w_j) = f(w_i, w_j) \quad (1)$$

Where $f(w_i, w_j)$ is the frequency of occurrences of a pair of words in a sentence.

The distance score is taken from the co-occurrence matrix. The weight $ew(w_i, w_j)$ of an edge can be calculated by equation 2.

$$ew(w_i, w_j) = \frac{1}{d(w_i, w_j) + smooth} \quad (2)$$

Where $d(w_i, w_j)$ is the distance score between two words on the graph. The smooth value is a factor added to avoid the division by zero because the distance score may be zero sometimes. The default value of smoothing is 0.1.

3.3 Word Clustering Algorithm

To generate a set of word clusters, this component represents a set of occurrence words on the undirected weighted co-occurrence graph where nodes represent words, and edges represent relations between them based on the co-occurrence statistics of words. This component uses two main phases in the process of the construction of word clusters. In the first phase, a set of word clusters is constructed and updated using agglomerative hierarchical and K-means clustering. For the second phase, the small clusters and isolated words are merged into another cluster. This component is illustrated in Figure 3.

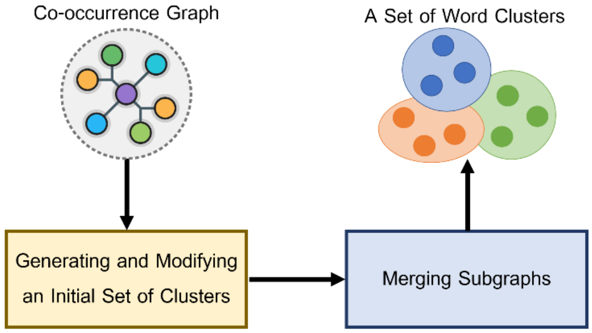

Figure 3. The word clustering algorithm

This component starts generating and modifying an initial set of word clusters on the undirected weighted co-occurrence graph. Then, the phase of merging subgraphs is processed for reducing in small clusters and isolated words.

3.3.1 Generating an initial set of word clusters

The main objective of this phase is to generate and modify an initial set of clusters in the undirected weighted co-occurrence graph. The algorithm first adapts hierarchical agglomerative clustering and K-means algorithm to cluster words into clusters by calculating the distance between words with finding the shortest path in the graph. The reason for combining two algorithms with the average-linkage strategy is to handle noisy text and outliers. Moreover, this technique does not require initializing clusters with centroids because it starts with one cluster. Initially, each word is taken as a cluster. Then, the closest pair of clusters are merged to form a bigger cluster. The average-linkage strategy is used to compute the similarity between the current cluster and all other candidate clusters. Then, the cluster having the highest similarity is identified for grouping. A recursive procedure is followed to build the complete tree according to a value of distance range until convergence criteria or no words moved to groups. For calculating the distance between words in the graph, Dijkstra's algorithm, the most common solution in graph theory, is utilized for finding the shortest path between two words in a graph. The relationship between words is calculated as a distance that indicates the cost of traversing between a pair of words. The distance from the source to a destination is calculated as a traversing cost between a pair of words. The total of the lowest cost is selected as a distance between a pair of words in the graph. The algorithm partitions the words into several clusters by iteratively updating the cluster centers and the associated words. The process of constructing and modifying an initial set of word clusters has the following steps.

1. In the initial step, each word is taken as an individual cluster.
2. The closest pair of clusters are merged when the distance between two clusters has less than the threshold of the distance range. The word is greater than or equal to the distance range is selected as a new cluster, and formed as a single cluster.

3. Compute the cluster center (centroid) based on the minimum average distance of all words in the cluster. To calculate a centroid, the centroid of a document is the word with the minimum average distance to all words of the document in the co-occurrence graph [16].

4. Assign the cluster to the closest centroid and according to the threshold of the distance range. In case of the distance between the incoming cluster and the centroid has less than the threshold of the distance range, the cluster is added into the cluster that has the closest centroid. Otherwise, the cluster is built as a new cluster.

5. Recompute (update) a new centroid based on the mean of all words in the cluster.

6. Calculate the average distance and distance range.

7. Repeat 4 to 6 steps until the mean of the clusters stops changing (do not change), or the criterion function converges, or the maximum number of iterations is reached.

After finishing this phase, an initial set of word clusters is generated. However, some initial word clusters, including small clusters and isolated words, appearing in the result of word clusters. To merge the small clusters and isolated words, the next phase is processed.

3.3.2 Merging subgraphs

The main idea of this phase is to post-process the initial word clusters, which have small clusters and isolated words. The objective aims to reduce the number of clusters and merge similar clusters to become the same cluster. Each small cluster and isolated word are assigned to the closest cluster based on the minimum distance between the word and the centroid and according to the threshold of the distance range. In case of the distance between the word and the centroid is higher than a pre-defined threshold. The word is not merged and then creating a new cluster. The small clusters and isolated words are then iteratively merged until convergence criteria, or no words moved to groups (i.e., no words change clusters, the sum of the distances is minimized, or some maximum number of iterations is reached). Therefore, each cluster contains the most similar words.

4. PROCESS OF EXPERIMENTS

This section aims to evaluate the performance of the proposed method and compare the proposed method against other word clustering algorithms. The experimental setup and goals are explained. The experiment is presented focused on the quality of result clusters. The detail of the process of experiments is shown in subsections.

4.1 Dataset

To evaluate the effectiveness and efficiency of the proposed method, the dataset is randomly selected by humans consisting of 100 articles from Asian Geographic magazine in 2019. The articles include the topic of art, car, computer, leisure, and sport. These articles are then converted from the PDF files into the plain text files, and manually removed unnecessary parts like graphics, tables, and figures. Human judges manually annotate the dataset due to no public dataset suitable for word clustering evaluation. This dataset is then a ground truth for evaluating the performance of the word clustering algorithm against other word clustering algorithms.

4.2 Evaluation Metric

To compare the quality of word clusters, the purity measure is employed to evaluate the accuracy of the word clustering algorithm. Purity is an external evaluation measure that needs a ground truth dataset. To compute the purity, each cluster is assigned to the class, which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned words and dividing by the number of all words in the cluster. The purity that closes to 0 means a weak clustering, and a perfect clustering has a purity of 1. The calculation of purity is defined by equation 3 as follows.

$$Purity = \frac{1}{N}\sum_{i=1}^{k} max_j \left|c_i \cap t_j\right| \qquad (3)$$

Where N is the number of all words, k is the number of clusters, c_i is a cluster in a set of clusters, and t_j is the classification, which has the max count for a cluster c_i.

4.3 Experimental Setup

The experiments are focused on comparing the algorithms according to the quality of the clustering. To achieve the goal and efficiency of the proposed method, the empirical experiments conduct the proposed method compared against four classical clustering algorithms reported in the literature including K-means, K-means++, DBSCAN, and LSI, using scikit-learn library. The classical algorithms of clustering are defined to use default parameters with the same dataset. The ground truth dataset is considered to validate the effect on the purity measure. All the algorithms used in the experiments are implemented in the Java SE 8 platform. The experiments are performed on a personal computer (PC) with an Intel (R) Core (TM) i5-4570 at 3.20 GHz CPU with 8 GB DDR2 RAM, running Ubuntu 18.04 LTS.

5. RESULTS AND DISCUSSION

The experimental results of the proposed method against other word clustering algorithms are summarized in Table 1.

Table 1. Table captions should be placed above the table

Algorithms	Purity
Proposed method	0.383
K-means	0.104
K-means++	0.125
DBSCAN	0.115
LSI	0.117

From table 1, the evaluation results can be concluded that the proposed method yields a better performance of generating the quality of word clusters than the baseline algorithms. The proposed method has a better purity than the baseline methods on the same dataset because the proposed method adopts two clustering methods based on the graph-based model in the process of the word clusters construction. These algorithms can handle noisy text and outliers effectively and, moreover, the co-occurrence matrix and graph-based model can capture structural information in texts such as frequency of a word, the distance of a word, a sequence of words, and the left-right context of the word.

Table 2. The example results of word clusters generated by the proposed method

Cluster No.	Cluster membership
1	airport, provider, climb, speedboat, vessel, mist, drive, vehicle
2	pig, buffalo, deer, macaque, bird
3	rotu, daal, dessert, appetizer, spread, vegetable
4	leg, jaw, chin, ear, neck, scrubbing, stroke
5	islam, christianity, intermingling, manicheaenism, cult, tradition, anahita, mizrahi, dawn, folk, judaism, richness
6	seafood, ingredient, wellbeing, hero, cocktail, curry, selection, menu, salad, artisan, meat, buster

Table 2 shows the example results of word clusters generated by the proposed method. Several words in one cluster have a syntactic or semantic relationship to each other, which shows the validity of the proposed word clustering algorithms. Although the proposed method has the best performance among four clustering methods, the proposed method has some limitations in the quality of each cluster due to some unrelated words in the cluster, which leads to the quality of the cluster is quite faulty. The solution to deal with this problem is using named entity recognition to categorize key information (entities) in text. Entities can be names of people, organizations, locations, times, quantities, monetary values, and percentages.

On the same dataset, the experimental results are confirmed that the proposed method could achieve accuracy and performance in terms of the purity measure. The proposed method is a useful technique for unsupervised feature selection and dimension reduction, which reduces the amount of textual data. The key objective of this technique is to enhance accuracy and performance in various domains of natural language processing such as text summarization, information retrieval, and text classification.

6. CONCLUSIONS

This article proposes a novel method of unsupervised word clustering algorithm using a graph and centroid-based approach. The proposed method focuses on automatically grouping similar words into the same cluster. The proposed method applies the hierarchical agglomerative clustering and K-means algorithm concepts to find similar words according to the criterion of distance range on the co-occurrence graph. The output of the proposed method is a collection of clusters of words. The experimental results show that the proposed method consistently and significantly outperforms state-of-the-art baselines in word clustering algorithms on the ground truth dataset. This method is thus the appropriate tool for solving real-world applications various tasks of natural language processing and text mining in terms of unsupervised feature selection and dimension reduction. However, the proposed method effectively leverages statistical information of words from a text corpus that cannot capture the semantic relationship between words. For the future works, the research article plans to apply word embeddings to calculate the distance of words for clustering semantically similar words, and use a named entity recognition to identify named entities such as person names, organizations, locations.

7. REFERENCES

[1] Liu B., and Zhang L. 2012. A Survey of opinion mining and sentiment analysis. *Mining Text Data*. Springer, Boston, MA.

[2] Jiguang, L., and Xiaofei, Z. 2014. Word clustering based on un-lp algorithm. In *Proceedings of the First AHA Workshop on Information Discovery in Text*, 25-30.

[3] Swetha G., Arvind S., Singh S.P., and Johri P. 2020. Word clustering on small corpuses. *Lecture Notes in Electrical Engineering*, Springer, Singapore.

[4] Wu Q., Ye Y., Ng M., Su H., and Huang J. 2010. Exploiting word cluster information for unsupervised feature selection. *PRICAI 2010: Trends in Artificial Intelligence, Springer*, Berlin, Heidelberg.

[5] Sven Martin, Jörg Liermann, and Hermann Ney. 1998. Algorithms for bigram and trigram word clustering. *Speech Commun. 24*, 1 (April 1, 1998), 19-37.

[6] S. Na, L. Xumin, and G. Yong. 2010. Research on k-means clustering algorithm: An improved k-means clustering algorithm. *Third International Symposium on Intelligent Information Technology and Security Informatics*, 63-67.

[7] A. Kapoor, and A. Singhal. 2017. A comparative study of K-Means, K-Means++ and Fuzzy C-Means clustering algorithms. *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*, 1-6.

[8] Sharma, L., and Ramya, P. 2013. A review on density-based clustering algorithms for very large datasets.

[9] A. Kontostathis. 2007. Essential dimensions of latent semantic indexing (LSI). *40th Annual Hawaii International Conference on System Sciences (HICSS'07)*.

[10] Mario Kubek, and Herwig Unger. 2016. Centroid terms as text representatives. In *Proceedings of the 2016 ACM Symposium on Document Engineering*, New York, NY, USA, 2016. Association for Computing Machinery, 99-102.

[11] Herwig Unger, and Mario Kubek. 2018. On evolving text centroids. In Recent Advances in Information and Communication Technology 2018, Cham, 2019. Springer International Publishing, 75-82.

[12] Yutaka Matsuo, Takeshi Sakaki, Kōki Uchiyama, and Mitsuru Ishizuka. 2006. Graph-based word clustering using a web search engine. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP '06)*. Association for Computational Linguistics, USA, 542-550.

[13] King-Ip Lin, Ravikumar Kondadadi. 2001. A word-based soft clustering algorithm for documents. *Computers and Their Applications*, 391-394.

[14] L. Yuan. 2019. A comparison of several word clustering models. In *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chengdu, China, 783-786.

[15] Nikoletta Bassiou, and Constantine Kotropoulos. 2011. Long distance bigram models applied to word clustering, *Pattern Recognition*, Volume 44, Issue 1, 145-15.

[16] Supaporn Simcharoen, and Herwig Unger. 2019. Dynamic clustering for segregation of co-occurrence graphs, *Fortschritt-Berichte VDI*, 53-71.