ISSN: 1803-3814

DOCUMENT CLUSTERING USING SELF-ORGANIZING MAPS: A MULTI-FEATURES LAYERED APPROACH

Muhammad Rafi, Muhammad Waqar, Hareem Ajaz, Umar Ayub, Muhammad Danish

National University of Computer and Emerging Sciences
Computer Science Department
ST-4 Sector 17-D, National Highway, Karachi
Pakistan
muhammad.rafi@nu.edu.pk

Abstract: *Cluster analysis of textual documents is a common technique for better filtering, navigation, understanding and comprehension of the large document collection. Document clustering is an autonomous method that separate out large heterogeneous document collection into smaller more homogeneous sub-collections called clusters. Self-organizing maps (SOM) is a type of artificial neural network (ANN) that can be used to perform autonomous self-organization of high dimension feature space into low-dimensional projections called maps. It is considered a good method to perform clustering as both requires unsupervised processing. In this paper, we proposed a SOM using multi-layer, multi-feature to cluster documents. The paper implements a SOM using four layers containing lexical terms, phrases and sequences in bottom layers respectively and combining all at the top layers. The documents are processed to extract these features to feed the SOM. The internal weights and interconnections between these layers features(neurons) automatically settle through iterations with a small learning rate to discover the actual clusters. We have performed extensive set of experiments on standard text mining datasets like: NEWS20, Reuters and WebKB with evaluation measures F-Measure and Purity. The evaluation gives encouraging results and outperforms some of the existing approaches. We conclude that SOM with multi-features (lexical terms, phrases and sequences) and multi-layers can be very effective in producing high quality clusters on large document collections.*

Keywords: *Document Clustering, Text Mining, Neural Network, Unsupervised Learning, Self-Organizing Maps, Layered Approach*

1 Introduction

Text is abundantly available on internet in a variety of forms. The textual documents generally contain rich human knowledge on different subjects. This exponential growth of documents in private and public repositories force researchers to dig meaningful and useful information form these documents. Clustering is an unsupervised machine learning method which has find an important application for managing such repositories in terms of better filtering, navigation, segregation, understanding and comprehension of these reserve in an autonomous way, clustering in this special scenario is specifically called document clustering. There are three main steps in document clustering, document representation, similarity function and clustering algorithm. Document clustering is very sensitive to representation scheme. Documents can say the same thing but using different vocabulary this is called vocabulary problem. Documents contain varying type of information even if they belong to the same genre [7, 15] and so their distribution in groups where similar documents are coupled closer together while dissimilar documents and their respective cluster is stored further apart is the challenge of document clustering. There are widespread use of documents in all fields like earth sciences, medicine, computer sciences, business math and more [7, 15]. Document clustering is an NP Hard problem [1] and has several challenges that need to be addressed to obtain a solution. The basic principle of Document Clustering is that it looks for patterns implicitly inside the textual data.

Firstly, we need to determine how many distinguishable groups of documents can and should be formed in a given collection and finding optimal assignments for each cluster. Second challenge is the vocabulary problem in which we have to cater sentences with same context but different usage of vocabulary.

Soft Computing approaches use a probabilistic criteria to attempt to solve problems that take too much time or computational resources. Document clustering is a non-deterministic polynomial (NP) Hard problem meanwhile we need an approach that utilize minimized time and computational resources and perform unsupervised clustering. Our solution should be robust and shouldn't require any training, hence minimizing the memory requirements.

1.1 Self Organizing Maps (SOM)

Self-organizing maps is a very popular neural network model approach in that can perform unsupervised clustering[8]. SOM works on unsupervised competitive learning where the winner neurons change the neural network to conform to its feature values. In other words the input changes the network to adjust (self-organize) itself. The SOM works on a layer where all feature vectors (neurons) are mapped. The number of centroids (clusters) in the neural networks and their initial place in SOM is chosen randomly or based on some error function. SOM is particularly useful as it takes comparatively very little computational resources and works quickly due to its nature as an iterative algorithm compared to some other unsupervised approaches, for example when compared to K-Means or HAC[8].
The biggest reason for using SOM in order to cluster our data set is because of its unsupervised approach. Unsupervised learning simply means that the input data is without labeled categories or classes, therefore SOM must determine by itself which groups to form completely by inferring from the data sets themselves. To achieve this goal we need to find a suitable representation for the documents. The motivation of using multiple layers with SOM lies in appropriate feature selection and weighting scheme. Large data sets have a huge number of documents with a lot of features. We need to limit number of features by selecting the most meaningful ones, in order to save time and computational resource. Multi layers will assist us to focus on features apart from words while words serving as our base layer. To create better clusters as well improving results we are using four layers instead of single SOM layer. Our four layers are going to consist of words (most frequent single term), phrases (most frequent bi-words in order), sequences (most frequent tri-words in order), and combined feature layer (frequent words + non-frequent words that are part of phrases and sequences). Each phrases should have at least one word that is present in the most frequent word list similarly each sequence should have at least one phrase present from the most frequent phrase list. The last layer is fed to the SOM engine. Our research will show that using multi layered approach instead of a single layer, we can improve our clustering.

2 Literature Review

We want to discuss relevant literature in this heading. [6, 14] proposes a method involving semantics to text document clustering to improve results using WordNet lexical in combination in SOM. The approach [2, 6] categorizes words into certain labels and works on it like the word cat and mouse both belong to the same category noun animal. Some words also has multiple categories like word Jaguar has 3 lexical categories (noun car, noun animal, noun company). Authors of paper [13, 14] have proposed conventional self-organizing map model (ConSOM) for text document clustering. ConSOM uses neurons or documents represented by two vectors. The first vector holds (tf*idf) score while the other semantic knowledge of that document. [14]. We understand the need to reduce the dimension space of the documents to favor faster execution. PCA (Principal Component Analysis) and LSI (Latent Semantic Indexing) are two widely used dimension reduction techniques used in literature [13, 12, 17, 21] to reduce the corpus size without impacting the results significantly. Another way to look at tokenization of documents is using Frequent max substring technique [5, 3, 19] to improve the efficiency of Information Retrieval. This is especially effective in certain Asian languages such as Chinese, Japanese, Korean and Thai. These languages are called non-segmented languages, i.e., a sentence is written continuously as a sequence of characters without explicit word boundary delimiters. Hence, sub-string or phrases as tokens are more appropriate. We looked at use of SOM in multi-language domain. This paper [2] suggests a novel approach of running SOM after a detailed pre-processing on Arabic Crime Documents. There were several challenges they encountered; and some were very relevant to us like Stemming and then conversion to another natural language or vice versa leads to bad conversion [2]. Multilanguage conversion of words using lemmatization without semantics is another bad conversion [2].
Clustering is the science of grouping a set of similar objects into one cluster, according to some criteria, and non-similar objects (and consequently clusters) are stored further apart. Clustering itself is not a specific algorithm, rather it can be achieved using various algorithms or approaches that differ based on our data set, how we define a cluster, and how to measure what we are looking among some other key questions. Clustering also requires setting certain parameters such as a function to calculate distance between objects, and number of expected clusters. Clustering approaches may differ based on a relationship of the clusters with each other, for example, a hierarchy of clusters embedded in each other (we can use Hierarchical Agglomerative Clustering) or that a cluster is prone to be shaped into circular formation (we can use K Means). Clustering is roughly distinguished as hard or soft clustering. Hard clustering refers to an approach where a document must belong to a single cluster only whereas in Soft clustering also known as fuzzy clustering each document may belong to N number of clusters. This situation can be explained in example where we have a news article about a politician being shot dead during a local sports match. The issue arises that as the story published in the news will mostly be about who the politician was and his lifes work, comments of people close to him and details of how he was killed. The resulting algorithm will put this document into cluster of deaths, obituary, or should it be in cluster

with local news, political news, or sports, rather this article belongs to each cluster of the a above categories and more to certain degree.

Document clustering are divided into two main approaches; hierarchical or partitive clustering. These are certain distinct features of both approaches that lead to their results and their use. In some cases, one approach is better whereas in some other cases it fails. Hierarchical approach is implemented using either agglomerative (bottom-up) or divisive (top-down) approach in order to build a tree structure. Agglomerative approaches are more common. We record each step of the HAC algorithm in a tree like structure called a dendrogram. In the dendrogram the leaf nodes are clusters containing one document only and the levels further upwards in the dendrogram consist of lesser clusters (merged clusters). We can pick whichever level of the dendrogram we need to satisfy our requirements. Partitive clustering works on separating the dataset into a number of disjoint groups. This is usually done by appointing each document to a K number of clusters. The K or number of clusters is usually predefined or can be a part of some error function. We have seen in paper [4] Multilayer self-organizing map (MLSOM) is featured in document retrieval (DR) and plagiarism detection (PD) system by modeling a rich tree-structured representation which efficiently retrieve a full document against a query. Another form of SOM is also used by forming Multi-layer Hierarchy which forms arbitrary complex clusters based on unsupervised method by establishing multilayer feedforward network, it records and compares the distance between clusters and given points of documents and also weight the points for determining where they belong [10].

Self-Organizing Maps or Kohonen Self-Organizing Maps are a type of artificial neural network which work no supervision is required [7]. The SOM architecture learns on its own through unsupervised competitive learning. The SOM engine takes in neurons of input data and the SOM clustering layer Maps the neuron based on its weight onto the layer. As more and more neurons are entered into the SOM, the network adapts itself to become like the input.

3 Document Clustering Using Self Organizing Maps:Multi-Layered Feature Space Approach

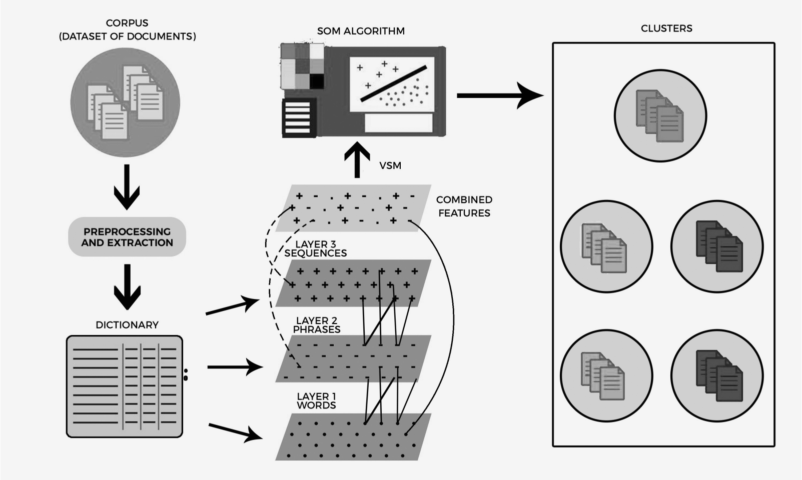

Figure 1: Proposed Approach

Our Approach combine SOM with Layers to improve cluster results. Traditionally, a SOM algorithm takes neurons and maps these neurons to centroids that match the neuron (winning neuron). Our neurons consist of feature vectors. We choose a certain number of features chosen based on being impactful in terms of occurrences in the collection. These neurons or feature vectors are then fed into the SOM algorithm that have more intelligent neurons and hence our results are seen to improve. Our algorithm is divided into two steps namely preprocessing step and final clustering approach using SOM.

3.1 Preprocessing of Documents

In the preprocessing step, the documents are processed and we extract multiple types of features from the corpus. Calculation of feature score and formation of layered architecture is the main intent of this step.

Selection and Score

Feature score is the value of tf*idf for each feature. Tf is the term frequency i.e. how many times a word has appeared in a particular document. Df is the document frequency of a feature i.e. in how many documents a feature has appeared. Inverted document frequency idf is the normalized form of document frequency(df). We have used logarithmic normalization for normalizing document frequency.

$$tfidf(d,t) = tf(d,t) \times \frac{\log |D|}{df(t)} \tag{1}$$

Layered Architecture

Layer 1 : The bottom layer consist of frequent words based on their occurrences in the corpus.

Layer 2 : The second layer consist of frequent phrases (in-ordered bi-words) based on their occurrences in the corpus, provided each phrase in this layer has atleast one word form layer 1.

Layer 3 : The third layer consist of frequent sequences (in-ordered tri-words) based on their occurrences in the document, provided each sequence has atleast one phrase from layer 2.

Layer 4 : The top most layer consist of words from layer 1 and words from phrases and sequences that are not present in layer 1.

Layer 4 is the input for the SOM algorithm. A vector space model of features(Layer 4) is created, where each row represents a document vector containing the feature scores.

Consider the following example:

Doc 1 : I like playing football in rain.

Doc 2 : It is going to rain today.

Doc 3 : John like to play football

*stop words are eliminated during the extraction process.

let say we have selected top 3 words, phrases and sequences based on occurrence in the collection. **Layer 1**

Frequent words : <football , rain , like>

Layer 2

Frequent Phrases : <like playing , like play ,playing football>

Layer 3

Frequent Sequences : <like playing football , like play football , playing football rain>

Layer 4

Combined Features : <football , rain , like , playing , play>

```
Data: Collection of Documents
Result: Vector Space Model of features
initialization;
$i \leftarrow 0$ ;
while $i <$ Total number of documents in the collection do
    while Di has next word available do
        read Di;
        tokenize Di ;
        remove special characters ;
        remove stop-words ;
    end
    form phrases and sequences from tokens ;
    calculate tf of each token, phrase , sequence ;
    Increment i ;
end
$R^n \leftarrow$ frequent N words from the collection ;
$R^m \leftarrow$ frequent M phrases from the collection ;
$R^l \leftarrow$ frequent N sequences from the collection ;
calculate tf*idf score of $R^n, R^m$ and $R^l$ as in equation 1;
$R^c \leftarrow R^n +$ non frequent words from phrases and sequences ;
create vector space model for $R^c$;
```

Algorithm 1: Preprocessor Algorithm

Let R^m be m dimensional frequent words , R^n be n dimensional frequent phrases and R^l be l dimensional frequent sequences then combined features will be $R^{m+(n-m)+(l-m)}$. Also l < n < m.

3.2 Clustering using SOM

The second step is SOM algorithm where the actual clustering task is done. The algorithm takes vector space model (generated from the preprocessing step) as input. The algorithm randomly picks specified number of initial seeds(document vectors) from the vector space model (VSM) to start processing. Distance of each document is calculated from the centroids and the document is associated with the closest centroid. The distance is calculated using equation 2 where d_i is the centroid d_j is the document vector whose distance from the centroid is to calculated. The winner centroid is updated , taking the average between the vectors of the document and the winner centroids, in this way atleast one centroid is updated on each iteration. In the next iteration, the next document vector gets a new set of centroids. There are two stopping criteria for our algorithm, either the provided number of epochs are completed or the clustering results of the current iteration are same as the previous iteration.

$$\cos(d_i, d_j) = \frac{d_i.d_j}{||d_i||^2||d_j||^2} \tag{2}$$

```
Start;
Select output layer network topology. ;
Select random seeds from the given document vectors;
$t \leftarrow 0$ ;
while stopping criteria do
    Select an input sample Di;
    Compute the cosine distance of Di to output weight vector;
    Attach Di with the seed with minimum distance;
    Select output winner node j that has weight vectors with minimum values;
    Update weights to all nodes within a topological distance;
    Increment t;
end
```

Algorithm 2: Clustering using SOM Algorithm

4 Experimental study

We are using Reuters, WebKB, and Doc 50 which are all renowned data sets and ideal for testing text clustering algorithms.

4.1 Data Set and Dimension Settings

Table 1: Data sets statistics

Corpus	Documents	Words	Phrases	Sequences
Doc50	50	6262	15760	17296
Re1 (Subset of Reuters)	216	10629	21248	24136
Re2 (Subset of Reuters)	650	20653	49362	57377
Re3 (Subset of Reuters)	845	25832	64994	75913
Course (Subset of WebKB)	50	3589	8033	9122
Project (Subset of WebKB)	100	7024	17811	20000

Table 2: Dimension Settings

Dimension	Words	Phrases	Sequences
D1	500	300	200
D2	1000	800	500
D3	1500	1000	500

We have selected these dimensions specifically instead of working on all the features because we wanted to only work on the frequently occurring features in the collection. This method focuses on significant features only. Moreover, it reduces the weight calculation complexity as the number of features are decreased. Analyzing the complexity of the dimensions, we have complexity of D1 < complexity of D2 < complexity of D3 because of calculation of feature score.

4.2 Evaluation Criteria

4.2.1 Purity

Purity is a simple and transparent evaluation measure. It can be defined as the maximal precision value for each class j. The cluster purity indicates the percentage of the dominant class members in the given cluster. The overall purity of the cluster C, can be computed as the weighted average purity. To compute purity using 3, 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 documents and dividing by N [11, 9, 20].

$$Purity(\omega, C) = \frac{1}{N} \sum_{k}^{maxj} |w_k \cap c_j| \tag{3}$$

4.2.2 F-measure

F-measure combines the precision and recall ideas from information retrieval[11, 18, 9]. We treat each cluster as if it were the result of a query and each cluster as if it were the desired set of documents for a query. We then calculate the recall and precision of that cluster for each given class using equation4 where P is the precision and R is the recall.

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{4}$$

We have used pair-wise precision and recall in calculation of F-measure.

4.3 Comparison with other approaches

For comparison with our approach, we implemented k-mean, harmony k-mean algorithms using the background knowledge proposed in [16] and single feature layer SOM. These algorithms work on single layers, forming vector space model of words extracted from the documents. The stop words were removed from the word list and feature score in the vector space model was same as ours i.e. tf*idf. The parameters we provided were same as the ones provided to our algorithm, no of clusters and dimension settings.

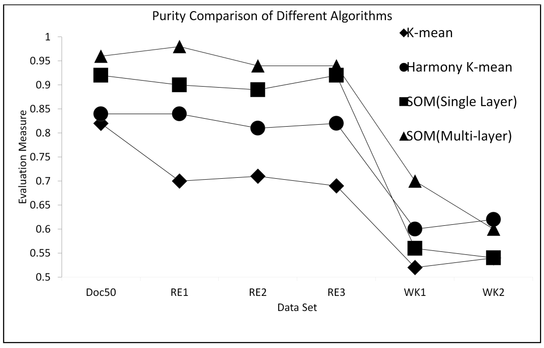

Figure 2: Comparison of purity of multi-layer SOM with other approaches

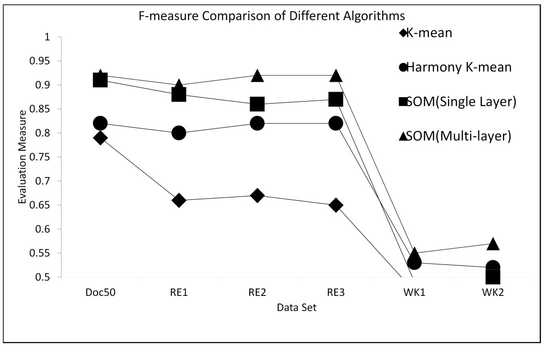

Figure 3: Comparison of F-measure of multi-layer SOM with other approaches

5 Results and Discussion

For details about dimension settings D1, D2 and D3 refer to table 2. The results in table 3 depicts that purity and F-measure of RE1, RE2 and RE3 is consistent on D1. This shows that smaller dimension contains significant features that are helpful in differentiating between the documents. After testing on the subsets of Reuter's data set, we can interpret that on complete data set of Reuter, smaller dimensions will produce better result. The purity and F-measure refer to tables 3 and 4 of both(Course, Project) data sets of WebKB are consistent with minor difference on all the dimensions. Therefore, we can take smaller dimension to decrease the feature score calculation complexity. Doc50 is an overly simplified testing data set for clustering. The purity and F-measure of D2 and D3 are consistent and giving better results.

Table 3: Purity evaluation of all data sets on D1 D2 D3

Dimension	Re1	Re2	Re3	WebKB(course)	WebKB(project)	Doc50
D1	0.9846	0.9461	0.942	0.66	0.59	0.94
D2	0.9523	0.8861	0.8852	0.7	0.5801	0.96
D3	0.9615	0.8861	0.8816	0.68	0.6	0.96

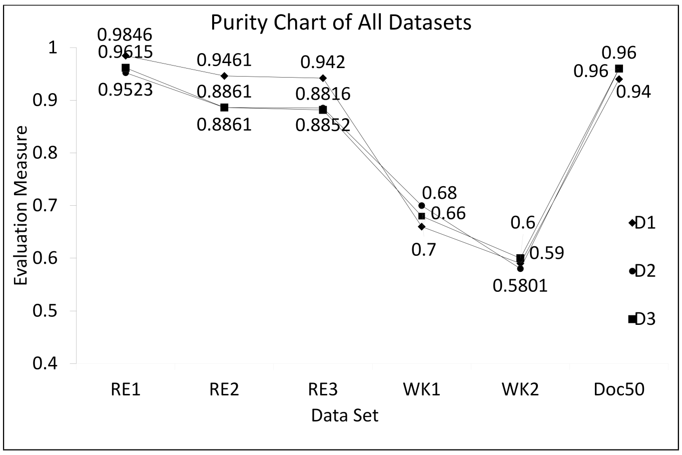

Figure 4: Purity evaluation of all data sets on D1 D2 D3

Table 4: F-measure evaluation of all data sets on D1 D2 D3

Dimension	Re1	Re2	Re3	WebKB(course)	WebKB(project)	Doc50
D1	0.9694	0.9057	0.9015	0.5366	0.4242	0.8824
D2	0.9215	0.8777	0.8824	0.5584	0.4404	0.9159
D3	0.9306	0.8772	0.882	0.5398	0.4261	0.9183

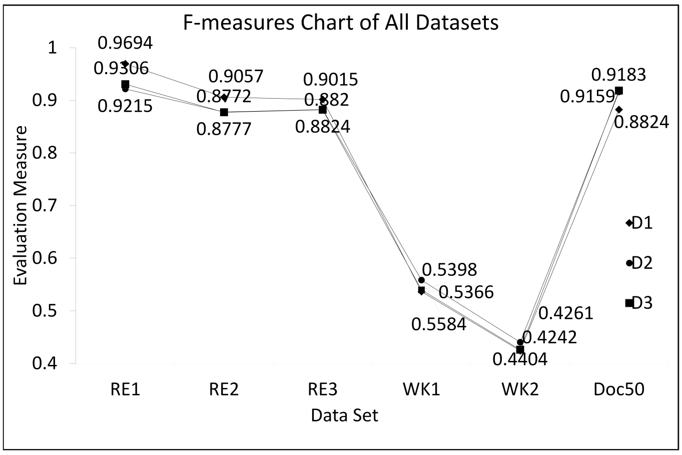

Figure 5: F-measure evaluation of all data sets on D1 D2 D3

6 Conclusion

The multi-feature layer approach using SOM can be very effective for clustering large document collection using a handful of features like words, phrases and sequences.

References

[1] Ajith Abraham, Swagatam Das, and Amit Konar. Document clustering using differential evolution. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 1784–1791. IEEE, (2006).

[2] Meshrif Alruily, Aladdin Ayesh, and Abdulsamad Al-Marghilani. Using self organizing map to cluster arabic crime documents. In *Computer Science and Information Technology (IMCSIT), Proceedings of the 2010 International Multiconference on*, pages 357–363. IEEE, (2010).

[3] Joachim Buhmann and Hans Kühnel. Complexity optimized data clustering by competitive neural networks. *Neural Computation*, 5(1):75–88, (1993).

[4] Tommy WS Chow and MKM Rahman. Multilayer som with tree-structured data for efficient document retrieval and plagiarism detection. *IEEE Transactions on Neural Networks*, 20(9):1385–1402, (2009).

[5] Todsanai Chumwatana, Kok Wai Wong, Hong Xie, et al. A som-based document clustering using frequent max substrings for non-segmented texts. *Journal of Intelligent Learning Systems and Applications*, 2(03):117, (2010).

[6] Tarek F Gharib, Mohammed M Fouad, Abdulfattah Mashat, and Ibrahim Bidawi. Self organizing map-based document clustering using wordnet ontologies. *IJCSI International Journal of Computer Science Issues*, 9(1):1694–0814, (2012).

[7] Shyam M Guthikonda. Kohonen self-organizing maps. *Wittenberg University*, (2005).

[8] Dino Isa, VP Kallimani, and Lam Hong Lee. Using the self organizing map for clustering of text documents. *Expert Systems with Applications*, 36(5):9584–9591, (2009).

[9] Gerald Kowalski. Information retrieval systems: theory and implementation. *Computers and Mathematics with Applications*, 5(35):133, (1998).

[10] Jouko Lampinen and Erkki Oja. Clustering properties of hierarchical self-organizing maps. In *Mathematical Nonlinear Image Processing*, pages 165–176. Springer, (1993).

[11] Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–22. ACM, (1999).

[12] Kristina Lerman. Document clustering in reduced dimension vector space. *Unpublished Manuscript*, (1999).

[13] Yuan-Chao Liu, Ming Liu, and Xiao-Long Wang. *Application of self-organizing maps in text clustering: a review*. INTECH Open Access Publisher, (2012).

[14] Yuanchao Liu, Xiaolong Wang, and Chong Wu. Consom: A conceptional self-organizing map model for text clustering. *Neurocomputing*, 71(4):857–862, (2008).

[15] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, (2008).

[16] Muhammad Rafi, Sufyan Shahid, Junaid Aftab, Muhammad Faizan Uddin, and Muhammad Shahid Shaikh. Towards a soft computing approach to document clustering. In *Proceedings of the 2017 International Conference on Machine Learning and Soft Computing*, pages 74–81. ACM, (2017).

[17] Osiski S. Dimensionality reduction techniques for search results clustering. Master's thesis, University of Sheffield, UK, (2004).

[18] Hinrich Schütze and Craig Silverstein. Projections for efficient document clustering. In *ACM SIGIR Forum*, volume 31, pages 74–81. ACM, (1997).

[19] Michael Steinbach, George Karypis, Vipin Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, (2000).

[20] Cornelis Joost Van Rijsbergen. Information retrieval. (1979).

[21] Bill B Wang, Robert I Mckay, Hussein A Abbass, and Michael Barlow. A comparative study for domain ontology guided feature extraction. In *Proceedings of the 26th Australasian computer science conference-Volume 16*, pages 69–78. Australian Computer Society, Inc., (2003).