A Fuzzy Document Clustering Model Based on Relevant Ranked Terms

K. Sreelekshmi and R. Remya

Abstract The web today is a growing universe of vast amounts of documents. Clustering techniques help to enhance information retrieval and processing huge volume of data, as it groups similar documents into one group. The relevant feature identification from a high-dimensional data is one of the challenges in text document clustering. We propose a sentence ranking approach which finds out the relevant terms in the documents so as to improve the feature identification and selection. Preserving the correlation between terms in the document, the document vectors are mapped into a lower dimensional concept space. We used k-rank approximation method which minimizes the error between the original term-document matrix and its map in the concept space. The similarity matrix is converted into a fuzzy equivalence relation by calculating the max-min transitive closure. On this, we applied fuzzy rules to efficiently cluster the documents. Our proposed method has shown good accuracy than previously known techniques.

Keywords Feature identification · Sentence ranking · Dimension reduction
Fuzzy document clustering

1 Introduction

A knowledge repository typically has vast and huge amounts of formal data elements, which are generally available as documents. With the rapid growth of text documents in knowledge repositories over time, textual data have become high-dimensional,

K. Sreelekshmi (✉) · R. Remya
Department of Computer Science and Engineering, Amrita School of Engineering, Amritapuri, India
e-mail: sreewarrier20@gmail.com

R. Remya
e-mail: remyar@am.amrita.edu

K. Sreelekshmi · R. Remya
Amrita Vishwa Vidyapeetham Amrita University, Coimbatore, India

© Springer Nature Singapore Pte Ltd. 2018

P. K. Sa et al. (eds.), *Recent Findings in Intelligent Computing Techniques*,
Advances in Intelligent Systems and Computing 709,
https://doi.org/10.1007/978-981-10-8633-5_11

which increases the processing time, and thereby diminishing the performance of the system. Thus, effective management of this ever-increasing volume of documents is essential for fast information retrieval, browsing, sharing, and comprehension. Text clustering is useful in organizing large document collections into smaller meaningful groups. Clustering aims at grouping objects having similar properties into the same group and others into separate groups based on the information patterns enclosed in it [1–4]. Numerous methods have been proposed and implemented to solve the clustering problem. The clustered documents will have highest similarity to the ones in the same cluster and least similarity to the ones in the other clusters.

According to the literature, clustering algorithms use either top-down or bottom-up approaches. Those which use top-down algorithms [5–7] work with the value of k which needs to be fixed and known in advance. Eventhough bottom-up approaches are effective, they lack performance in certain datasets. Recently, several types of biologically inspired algorithms have been proposed for clustering [8].

Despite of all these approaches, however, document clustering still presents certain challenges. This includes optimizing feature selection for document representation, reduction of this representation into a lower dimensional pseudo-space, with less information loss and incorporating mutual information between the documents into a clustering algorithm. Our work proposes an approach which extracts the relevant terms by a sentence ranking approach. The work also proposes a dynamic rank reduction method to map the documents into a low dimensional concept space. Finally, using the similarity measures, documents are clustered based on fuzzy rules.

2 Related Work

A method proposed in [9] uses sentence ranking and clustering based summarization approach to extract essential sentences from the document. A weighted undirected graph is constructed according to the order of sentences in a document, to discover central sentences. The weights are assigned to edges by using sentence similarity and discourse relationship between sentences. The scores of sentences are obtained by a graph ranking algorithm. Sentences in the document are clustered by using a Sparse Nonnegative Matrix Factorization. Tian et al. [10] use a page ranking approach to rank the sentences in a document. Li et al. [11] perform constrained reinforcements on a sentence graph, which unifies previous and current documents to determine importance of sentences. The constraints ensure that the most important sentences in current documents are updates to previous documents.

In [12], the authors proposed a fuzzy clustering framework which uses statistical measures to form sentence clusters. Related sentences are grouped by applying an enhanced fuzzy clustering algorithm. Semantically similar sentences are identified using Expectation Maximization (EM) framework and Page Rank score of an object. The work [13] focuses mainly on three issues, namely, optimizing feature selection, a low-dimensional document representation, and then incorporating these information into a clustering algorithm. To address the issue of feature extraction,

it considers a domain-specific ontology that provides a controlled vocabulary. The original feature space is mapped into a lower dimensional concept space, using Singular Value Decomposition (SVD). The relationship between documents is modeled using the fuzzy compatibility relation. With the help of a cluster validation index, all the data sequences are allocated into clusters. The drawback of the work is that it is restricted only to a particular domain with a controlled vocabulary. A fuzzy controlled genetic algorithm, combined with semantic similarity measure is used for text clustering in [14]. In this work, a hybrid model that combines the thesaurus based and the Semantic Space Model (SSM) approach is proposed. The biologically inspired principles of genetic algorithm with the help of fuzzy controllers significantly improved clustering accuracy.

Even though all these works address the problem with feature identification and extraction, accuracy still remains as an open problem to be considered. In our work, features are identified by a sentence ranking approach. Rather than fixing the dimension using a cluster validation index, we proposed a dynamic dimension-fixing approach which retains almost all of the information content (i.e., minimizes the error). We build a fuzzy controlled system where the clusters are determined automatically, based on the information patterns formed.

3 Proposed Approach

The work mainly focuses on three major phases in text document clustering.

1. Relevant feature selection and extraction.
2. Document representation so as to retrieve mutual information between documents.
3. Efficient clustering method.

Solution approach is shown in Fig. 1. Each module is explained in detail in the following subsections.

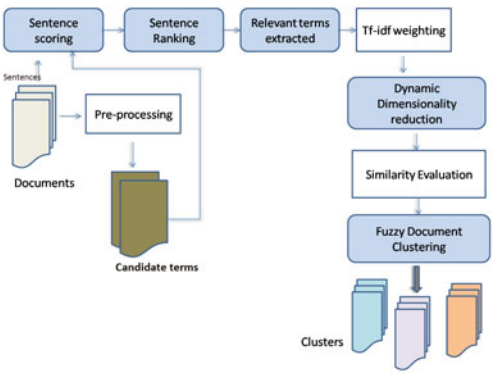


Fig. 1 Block diagram of the proposed work

Preprocessing of the data is done by stop word removal, lemmatization, and specific pattern removal to obtain the candidate terms. They are then written along with its frequencies to a temporary file, which is fed to the proposed system for further processing and clustering.

3.1 Feature Identification and Extraction

Feature identification is a challenging task in document clustering. A method to efficiently acquire the relevant terms in a document is one of the motives of the work. To reveal the relevance of the terms in the context, we put forth a method, which considers the *importance value* of sentences. For each document, we build a sentence vocabulary by considering the terms in the temporary file and the sentences in that document. Here, we introduce a new weighting measure to the terms, as the product of term frequency in a particular sentence and logarithmic ratio of total sentences to the number of sentences in which the term appears. Using (1), we obtain the weight of each term, where tf and sf represent the frequency of the term in the particular sentence and number of sentences in which the term appears, respectively.

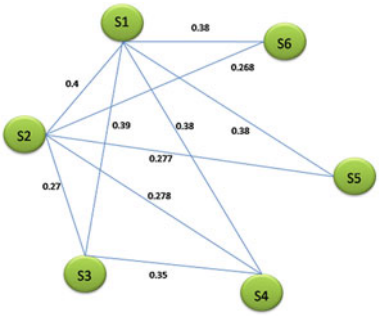
$$W_{term} = tf_{term}.\log_{10}\frac{N}{sf} \tag{1}$$

In the news corpus, introduction sentences convey the entire information in the article, and thus will have more significance compared to the sentences that follows. To accommodate this feature, we assign positional weights to the sentences in our document collection. The relationship between sentences is modeled as an undirected graph with sentences as nodes and, edges show which all sentences are related to each other. The relationship between sentences are found by taking advantage of the discourse relations [9], which reveals how much a sentence is dependent on another, by context. There is a set of predefined discourse terms. Let s_i and s_j be the two sentences where, s_j follows s_i in sequence. If s_j starts with a discourse term, then s_j is dependent on s_i. Thus, s_j imposes a weight on to s_i. This results in an increase in the weight of s_i. The necessary condition for two sentences to have a discourse relationship is that the sentences s_i and s_j should be adjacent. The cosine similarities between the sentences are also considered to improve the edge weights, by considering each sentences as vectors. We then combine all these measures to find the effective edge weights.

$$W_{effective} = a.W_{VSM} + b.W_{dis} + (1 - a - b).W_{pos} \tag{2}$$

where a and b are the weights. We considered the discourse relation has more relevance in revealing the relationship between sentences than cosine similarity, and are fixed to be 0.3 and 0.5, respectively. Thus, we get a fully connected weighted graph. An example is shown in (Fig. 2). Here, s_1 to s_6 are the sentences and edges

Fig. 2 Sentence weighting



show the weights assigned. The weights are normalized and given to the sentence ranking module. We then ranked the sentences by the Page Ranking Algorithm [9]. Thus, the top ranked k sentences are retrieved. With the assumption that all the terms in the important sentences will also be important, the relevant terms are extracted from the top ranked sentences. This is our relevant term collection. We then used the traditional tf-idf scoring to assign weights to the retrieved relevant terms in the document. Let G be the graph with E and V, the edges and vertices, respectively. Sentences are modeled as nodes of the graph. The algorithm returns the weights for all the sentences in the document.

Input : G(E,V): The graph
E : set of all edges.
V : set of all vertices.
N : Total number of lines in Di.
K : A multiplication constant.
Output : Edge Weights.

Algorithm 1 Edge weighting

for each i ϵ 1...N :
for each j ϵ 1...N :
if i==j :
set W_{ij}=0
else :
foreach : e_{ij} ϵ G(E,V) // for i,j ϵ 1...N
Wpos(S_i)=N*K/Line number
if $S_j[0] \in$ 'discourse terms' && i and j are adjacent :
Improve the weight W_{ij}.
endIf
endFor
endIf
endFor
endFor
Normalize the weight: $W_{ij} = W_{ij} / \sum_{i=1}^{n} W_{ik}$. //k: number of edges from node i.
return weight.

Input : G(E,V): The graph
V : set of all vertices(sentences).
E : set of all weighted edges(relationship between sentences).
Output : Rank of all sentences.

Algorithm 2 Sentence Ranking

```
for each sentence S_i ϵ doc D_i :                    // for i ϵ 1...N
add v_i to G.V
for each sentence S_i :                              iϵ 1...N:
for each sentence S_j :                              jϵ 1...N:
if Si ∼ Sj :
Add an undirected edge e_ij to G.E.
Add weight w_ij to the edge e_ij.
endIf
Compute rank of v_i
endFor
endFor
R(v_i)= d.∑ R_j x w_ij+(1 - d)                       //d: constant
endFor
return rank, R
```

3.2 Dimensionality Reduction

The term-document matrix, having the tf-idf score, is fed to the decomposition module. The original term-document matrix is mapped to a low dimensional concept space, using SVD. The drawback in using SVD for decomposition is that the reduced rank, k is fixed by inspection of the singular values. In the proposed method, k is considered dynamically by converting this into an energy conservation problem. Let λ_1, $\lambda_2 \ldots \lambda_n$ be the singular values [15]. The energy conserved for n-dimension will be,

$$E_n = \frac{\lambda_1^2 + \lambda_2^2 + \cdots + \lambda_i^2}{\lambda_1^2 + \lambda_2^2 + \cdots + \lambda_n^2} \tag{3}$$

where n is the total number of documents and i = 1, 2, …, n. Here, we consider 80% energy (information) is conserved. i.e., $E_n \geq 80\%$.
Input : A; *Term − document matrixofsize* $m \times n$.
Diagonal values of Sigma (λ_1,λ_2, … λ_n)
Output: Reduced Dimension.

Algorithm 3 Dimension Reduction

for each λ_i : // i = 1,2,...,n
if ${\lambda_i}^2/\sum_1^n {\lambda_i}^2 \leq 80\%$:
continue
else :
dimension=i
break
endIf
endFor
return dimension.

3.3 Similarity Evaluation

The document similarity is evaluated with the vector space model, in the reduced concept space. Each document is considered as a vector and the similarity is found [13].

3.4 Fuzzy Document Clustering

The fuzzy set theory is an extension of the classical set theory with degrees of membership for its elements [16]. In this work, we used the documents' similarity as the membership degrees to form a fuzzy equivalence relation. A fuzzy equivalence relation can be defined using the mathematical preliminaries defined in [13]. To cluster the documents, the following steps are done:

- To apply fuzzy rules, the similarity matrix is to be normalized to bring the values in the range [0, 1] .
- Considered max-min closure to find transitive closure R_T.
- Chose proper λ (cut-set) to find all feasible clusters.

Input : The similarity matrix R'
Output : Clusters.

Algorithm 4 Fuzzy Clustering

Perform a normalization computation on R'
R_{ij}=(R_{ij}'+1)/2
Find a transitive closure$R^T=R^{n-1}$
Find a suitable λ-cutset$\in$ [0,1]
for all R^T[i,j] $< \lambda$:
if i==j :
set R^T[i,j]=0
else :
set R^T[i,j]=1
endIf
Select the docs corresponding to 1's to fall in the same cluster
return clusters

4 Experimentation

To evaluate the proposed approach, we considered 406 documents from the 20-newsgroups collection, available in UCI Machine Learning Repository. The dataset includes documents from four different topics.

4.1 SStress Criteria

The dissimilarity between two matrices is given by the SStress criteria. SStress is defined as (4).

$$SStress = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(S_{ij}^2 - \widehat{S}_{ij}^2 \right) \tag{4}$$

S_{ij} and $\widehat{S}_{ij}$ refers to the elements in the original term-document matrix and the matrix in the reduced concept space, respectively. From Fig. 3, we can see that SStress decreases with the increase of dimension k. To fix an optimal dimension retaining 80% of the information, we fix the dimension at 247 which is shown in Fig. 4. With this dimension, we could map the original term-document matrix into a concept space without much information loss, thereby reducing the error.

4.2 Proper Cutoff for Clusters

Considering the accuracy of clusters formed against different λ-cut values, we chose a proper λ value by inspection. Figure 5 shows the plot of number of clusters versus

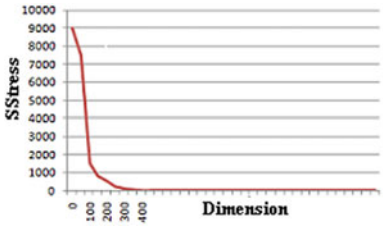


Fig. 3 The SStress between original matrix and the matrix in reduced concept space

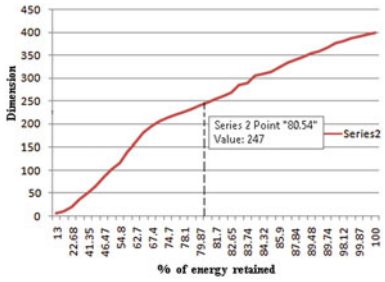


Fig. 4 Choosing optimal rank, k

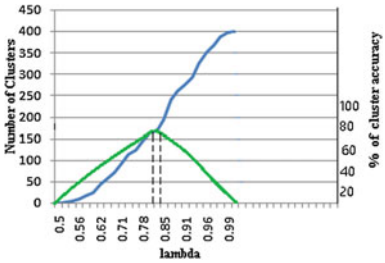


Fig. 5 Choosing a cutoff for fuzzy clustering

λ value for the dataset under consideration and we chose optimal λ value between 0.8 and 0.85 as shown.

5 Result and Analysis

We used the metrics; precision, recall and F1 score to compare our work with existing method (5).

$$Precision(P) = \frac{TP}{TP + FP}, Recall(R) = \frac{TP}{TP + FN}, F1Score = \frac{2.PR}{P + R} \quad (5)$$

TP, FP, and FN represents the true positive, false positive, and false negative rates, respectively. The proposed method correctly clustered majority of the documents. The confusion matrix is obtained as shown in Table 1. We used the selected subset of

Table 1 Confusion matrix (total number of documents = 406)

	Same class	Different class
Same cluster	300	40
Different cluster	36	30

Table 2 Performance comparison with different datasets

Dataset	Method	Precision	Recall	F-measure
Selected subset of 406 documents	Fuzzy clustering on domain-specified ontology [13]	0.77	0.82	0.79
	K-Means (WEKA tool)	0.668	0.60	0.632
	DB-Scan (WEKA tool)	0.72	0.75	0.73
	Fuzzy clustering based on relevant ranked terms	0.88	0.89	0.83
Document collection from [13]	Fuzzy clustering on domain-specified ontology [13]	0.78	0.80	0.78
	Fuzzy clustering based on relevant ranked terms	0.82	0.76	0.79

406 documents and those used in [13] to compare the performance of each method. Also, traditional approaches like K-means and DB-SCAN are run on our dataset using WEKA tool. Performance analysis of different methods is shown in Table 2.

6 Conclusion

We propose a method to select and extract relevant terms from documents by a sentence ranking approach. Also, a dynamic method has been proposed to choose a suitable rank in the lower dimensional space without losing relevant information. Fuzzy clustering applied to the similarity relation gives good quality clusters. From the experiment results obtained, it is evident that our system performs better than the traditional approaches and the one proposed in [13].

References

1. Gurrutxaga, I., Albisua, I., Arbelaitz, O., Martin, J.I., Muguerza, J., Perez, J.M., Perona, I.: An efficient method to find the best partition in hierarchical clustering based on a new cluster validity index. Pattern Recogn. **43**(10), 3364–3373 (2010)
2. Nguyen, C.D., Krzysztof, J.C.: GAKREM: a novel hybrid clustering algorithm. Inf. Sci. **178**, 4205–4227 (2008)
3. Saha, S., Bandyopadh yay, S.: A symmetry based multi objective clustering technique for automatic evolution of clusters. Pattern Recogn. **43**(3), 738–751 (2010)
4. Menon, R.R.K., Aswathi, P.: Document classification with hierarchically structured dictionaries. Adv. Intell. Syst. Comput. **385**, 387–397 (2016)
5. Selim, S., Ismail, M.: K-means-type algorithm: generalized convergence theorem and characterization of local optimality. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 81–87 (1984)
6. Bandyopadhyay, S., Maulik, U.: An evolutionary technique based on K-means algorithm for optimal clustering in R. Inf. Sci. **146**, 221–237 (2002)
7. Harikumar, S., Surya, P.V.: K-medoid clustering for heterogeneous datasets. Proc. Comput. Sci. **70**, 226–237 (2015)
8. Song, W., et al.: Genetic algorithm for text clustering using domain-specified ontology and evaluating various semantic similarity measures. Expert Syst. Appl. **36**, 9014–9095 (2009)
9. Ge, S.S., Zhang, Z., He, H.: Weighted graph model based sentence clustering and ranking for document summarization, Singapore National Research Foundation, Interactive Digital Media R&D Program, pp. 90–95 (2010)
10. Tian, J., et al.: Ranking sentences in scientific literatures. In: 11th International Conference on Semantics, Knowledge and Grids, pp. 275–282. IEEE (2015)
11. Li, Xuan, et al.: Update summarization via graph-based sentence ranking. IEEE Trans. Knowl. Data Eng. **25**(5), 1162–1174 (2013)
12. Uma Devi, M., et al.: An enhanced fuzzy clustering and expectation maximization framework based matching semantically similar sentences. In: 3rd International Conference on Recent Trends in Computing, Procedia Computer Science, vol. 57, pp. 1149–1159 (2015)
13. Yue, Lin, et al.: A fuzzy document clustering approach based on domain-specified ontology. Data Knowl. Eng. **100**, 148–166 (2015)
14. Song, Wei, et al.: Fuzzy control GA with a novel hybrid semantic similarity strategy for text clustering. Inf. Sci. **273**, 156–170 (2014)
15. Grewal, B.S.: Higher Engineering Mathematics. Khanna Publishers
16. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)