Chapter 4

Centroid-Based Lexical Clustering

Khaled Abdalgader

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.75433

Abstract

Conventional lexical-clustering algorithms treat text fragments as a mixed collection of words, with a semantic similarity between them calculated based on the term of how many the particular word occurs within the compared fragments. Whereas this technique is appropriate for clustering large-sized textual collections, it operates poorly when clustering small-sized texts such as sentences. This is due to compared sentences that may be linguistically similar despite having no words in common. This chapter presents a new version of the original *k*-means method for sentence-level text clustering that is relay on the idea of use of the related synonyms in order to construct the rich semantic vectors. These vectors represent a sentence using linguistic information resulting from a lexical database founded to determine the actual sense to a word, based on the context in which it occurs. Therefore, while traditional *k*-means method application is relay on calculating the distance between patterns, the new proposed version operates by calculating the semantic similarity between sentences. This allows it to capture a higher degree of semantic or linguistic information existing within the clustered sentences. Experimental results illustrate that the proposed version of clustering algorithm performs favorably against other well-known clustering algorithms on several standard datasets.

Keywords: semantic similarity, sentence-level text clustering, word sense identification, lexical resource

1. Introduction

Although lexical clustering at the document-level text is well studied in the natural language processing (NLP), computational linguistic, and knowledge discovery literature, clustering at the sentence-text level is challenged by the fact that word frequency—possible frequent occurrence of words from textual collection—on which most text semantic similarity methods are

based, may be absent between two semantically similar text fragments. To solve this problem, several sentence-level text similarity methods have recently been established [1–17][1].

The sentence similarity measures proposed by Li et al. [1], Mihalcea et al. [2], and Wang et al. [18] have two major features in common. Firstly, rather than using all possible features from applied external textual collections to representing sentences in a vector space model [19], only the words appearing in the compared sentences are used, thus solving the issue of data sparseness (i.e., high dimensionally) resulting from a randomly processing of the words (i.e., bag of words representation). Secondly, they use the available semantic and linguistic information from the applied lexical sources to solve the issue of deficiency of word co-occurrence.

The measures of sentence-level text similarity such as presented by Abdalgader and Skabar [10] (the latter of which we use in this chapter and described later in Section 2) depend in a way of using the word-related synonyms to calculating the semantic similarity between words. Unlike existing measure of short text semantic similarity, which use the exact words that appear in the compared sentences, this similarity method creates an expansion word set for each sentence using related synonyms of the sense-disambiguated words in that sentence. This way lead to provide a richer and highly connected semantic context to estimate sentence similarity through better utilization of the possible semantic information from the available lexical resources such as WordNet [20, 21]. For each of the sentences being calculated for their similarity, a *word sense identification* step is first applied in order to determine the correct sense based on the surrounding context [22]. A synonym expansion step is then applied, resulting in a richer and fully connected semantic context from which to estimate semantic vectors. The similarity between these vectors can then be calculated using a standard vector space similarity measure (i.e., cosine measure).

Several text-clustering methods: however, have been existed in the study [18, 23–37, 38–40, 42], and a majority of them consider the matrix of semantic similarities between words as input only. The *k*-medoids [30, 31] is one of these methods, which is considered as a developed version of *k*-means method in which centroids are restricted to being data patterns (i.e., points). However, a problem with the *k*-medoid method is that it is highly sensitive to the random selection (i.e., initial) of centroids, and in empirical executions, it is often requiring to be executed many times with different initialization settings. To solve this issue with *k*-medoids, Frey and Dueck [35] proposed *Affinity Propagation*, a graph-based algorithm that concurrently does take all data points as possible centroids (i.e., exemplars). Processing each data point as a node in a graph, affinity propagation recursively transfers real-valued messages along the vertices of the graph until a required set of possible centroids are achieved.

Another graph-based clustering method that depends on matrix decomposition techniques from the linear algebra theories is a spectral-clustering algorithm [18, 36, 37, 39, 41]. Rather than clustering data patterns in the traditional vector space model, it associated data patterns together with the space resulted from eigen-vectors linked with the top eigen-values and then apply clustering in this new transformed space, usually applying a *k*-means method. One of

[1]This chapter adapts the journal version that appeared in the IAENG International Journal of Computer Science, 44:4, IJCS_44_4_12 [42].

the benefits of this method is that it has the ability to classify non-convex classes, which is challenging when clustering by using *k*-means method (i.e., typical feature space). Since spectral-clustering method requires only a matrix comprising pairwise similarity as input, it is easy to apply it to the sentence-level text-clustering task [18, 29].

Erkan and Radev [43], Mihalcea and Tarau [44], and Fang et al. [46] have applied a PageRank [45] as a centrality measure in the task of document summarization, in which the aim is to rank sentences regarding their role in the document being summarized. Importantly, Skabar and Abdalgader [29] proposed a new fuzzy sentence-level text-clustering method that also uses PageRank as a centrality measure, and it allows clustered sentences to belong to all classes with different degrees of similarity (i.e., membership). The nation of this fuzzy clustering is required in the case of document summarization, in which a sentence may be linguistically similar or related to more than one topic [14, 29, 47].

The contribution presented in this chapter is a new version of the original *k*-means method for sentence-level text clustering that is dependent on the idea of using the related synonym sets to create rich and highly connected semantic vectors [42]. These vectors characterize sentence using semantic information derived from a WordNet to determine the actual sense to a word, based on the surrounding context. Thus, while the original *k*-means method is relay on calculating the distance between patterns, the new version is operating by calculating the semantic similarity between sentences. This allows it to capture more semantic information accessible within the clustered sentences. The result is a centroid-based lexical-clustering method which can be used in any application in which the relationship between patterns is expressed in terms of pairwise semantic similarities. We apply the algorithm to several benchmark datasets and compare its performance with that of well-known clustering methods such as *spectral clustering* [36], *affinity propagation* [35], *k-medoids* [30, 31], *STC-LE* [39], and *k-means (TF-IDF)* [40]). We claim that the satisfactory performance of new proposed version of the centroid-based lexical-clustering method is due to its ability to better utilize and capture a higher degree of semantic information available in used lexical resource.

The remainder of this chapter is organized as follows. Section 2 presents a representation scheme for calculating sentence semantic similarity. Section 3 describes the proposed variation of original *k*-means clustering (centroid-based) method. Empirical results are shown in Section 4, and Section 5 concludes the chapter.

2. Semantic similarity representation scheme

By far, the most widely used text representation scheme in the natural language processing activities is the vector space model (VSM), in which a text or a document is represented as a point in a high-dimensional (N_i) input space. Each dimension in this input space (i.e., VSM) corresponds to a unique word [19]. That is, a document d_j is represented as a vector $x_j = (word_{1j}, word_{2j}, word_{3j}, \ldots)$, where $word_{ij}$ is a weight that represents in some way the importance or relatedness of word $word_i$ in document d_j and is dependent on the frequency of occurrence of $word_i$ in document d_j. The semantic similarity between the compared documents is then

measured using the corresponding vectors, and a usually applied measure is the cosine of the angle between the two vectors.

The VSM has been effective in information retrieval (IR) activities because it is able to sufficiently utilize much of the semantic information expressed in the larger-sized textual collection. This is due to a large textual collection or documents may contain many shared words with each other and thus be considered similar regarding to well-known vector space similarity measures such as the cosine measure. However, in the case of sentence-level text (text fragment), this is not the case, since two sentences may be carrying the same meaning (i.e., semantically similar) whereas comprising no similar words. For instance, consider the sentences "*Some places in the country are now in torrent crisis*" and "*The current flood disaster affects the particular states.*" Obviously, these two sentences have the same meaning, yet the only common word they have is *the*, which does not carry any semantic information (i.e., stop words). The reason why word co-occurrence may be rare or even absent in sentences is due to the flexibility of natural language that allows humans to express the same meanings using very different sentences in terms of structure and length [50]. Therefore, we need a sentence-level text representation scheme which is superiorly able to utilize and capture all the possible semantic information of sentences, thus enabling a more efficient similarity method to be used.

2.1. Measuring sentence-level text similarity

To calculate the semantic similarity between two sentences, we use sentence similarity method that uses the sets of synonym expansion appeared in the compared sentences [10]. To demonstrate how this measure work: however, suppose that $Sentence_1$ and $Sentence_2$ are the two sentences being compared to calculate their semantic similarity, W_1 and W_2 are the sets of sense-assigned words appeared in $Sentence_1$ and $Sentence_2$, respectively, $sentence_1$ and $sentence_2$ are the sets of synonym expansion appeared in W_1 and W_2, and $U = W_1 \cup W_2$. Then, a semantic vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ have been created, according to $sentence_1$ and $sentence_2$.

Let $word_j$ be the corresponding sense-assigned word from U and v_{ij} be the j^{th} element of v_i. In this case, there are two instances to take into the account, relaying on whether $word_j$ appears in $sentence_i$ or not:

Instance 1: If $word_j$ exists in $sentence_i$, then set v_{ij} equal to the value of 1, this is based on the semantic similarity of the same words in the WordNet.

Instance 2: If $word_j$ does not exist in $sentence_i$, then compute the semantic similarity between compared words by using one of the WordNet-based word-to-word similarity measures (i.e., J&C measure) [51]. The final similarity score to v_{ij} is the highest of these scores between $word_j$ and each $sentence_i$.

Once the vectors ($\mathbf{v}_1$ and $\mathbf{v}_2$) have been constructed, the semantic similarity between two sentences can be determined using a cosine similarity measure between two constructed vectors as

$$Similarity(Sentence_1, Sentence_2) = (\mathbf{v}_1.\mathbf{v}_2)/(|\mathbf{v}_1\|\mathbf{v}_2|) \quad (1)$$

2.2. Sentence-level clustering algorithm

In this section, we firstly describe the new proposed version of the original *k*-means clustering algorithm which we called it centroid-based lexical-clustering (CBLC) algorithm. Then, we describe how a cluster centroid can be constructed and defined. The remaining subsections discuss the issues of calculating the semantic similarity between sentences and clustering centroid, and other related technical issues such as empirical settings and space and time complexity.

2.3. Centroid-based lexical clustering

Algorithm 1. Centroid-Based Lexical Clustering (CBLC).

Input: Sentences to be clustered $S = \{S_i \mid i = 1 \text{ to } TN\}$

Classes # k

Output: Membership values of each cluster $\left\{\pi_i^j | i = 1..TN, j = 1..k\right\}$ where π_i^j is the membership value of sentences i to cluster j.

1. *//Randomly distribute the sentences into k classes*
2. **for** $i = 1$ to TN
3. **if** $i \leq k$
4. j + =1
5. $\pi_i^j = S_i$ *//Sentence*$_i$
6. **else**
7. $j = 1$
8. $\pi_i^j = S_i$ *//Sentence*$_i$
9. **end**
10. **repeat until there is no move (until convergence)**
11. *//Define or determine the centroid for each class (cluster)*
12. **for** $j = 1$ to k
13. M_j = *union-set {all possible synonym occurring in the cluster* $_j$*} //U set*
14. **end**
15. *//Compute the similarity between each sentence* (S_i) *to each cluster centroid*
16. **for** $j = 1$ to k
17. *similarity*(M_j, S_m) // S_m *is sentences related to cluster j, {m = 1.. n}.*

18. end

19. *//Re-locate each sentence to the corresponding cluster centroid to which it is similar to.*

20. re-*locate*(S_i, M_j)

21. End

Given a *k* set (i.e., clusters), partition all the data points (i.e., sentences) randomly in given sets (i.e., initialization), each with a determined centroid (*mean*) that demonstrates as representative of the cluster. There are iterations process that rearrange these means or centroid of the clusters, which is based on moving each sentence to the cluster corresponding to the centroid to which it is closest (i.e., semantically similar). Redetermine the cluster centroids based on the new located sentences belonging to them. Then, the following iteration is repeated until the centroids do not move (until convergence). The new proposed version of the original *k*-means clustering algorithm is as follows.

2.4. Determining a clustering centroid

In the standard vector space model, the text such as a document is processed as a vector (i.e., its elements are the *tf-idf* scores), a cluster centroid can be determined by taking into account the vector average over all text fragments related to that cluster. This is experienced very hard using the above-discussed text representation scheme, since the semantic vector for a sentence is not unique, but depends on the length of the compared sentence context. However, just as a context may be constructed by two sentences, it is direct to apply this nation to defining the context over a collection of sentences. While a cluster is just such text fragments, we can define the centroid of a cluster as the *union set* of all associated synonyms of disambiguated words existing in the sentences relating to that cluster. Thus, if $Sentence_1$, $Sentence_2$, … $Sentence_N$ are sentences belonging to some cluster, the centroid of the cluster, which we denote as M_j, is just the union set {$word_1$, $word_2$, .. $word_n$}, where *n* is the number of distinct synonym words ($sentence_i$) in$Sentence_1 \cup Sentence_2 \cup \ldots \cup Sentence_N$. **Figure 1** exemplifies the idea of determining a clustering centroid.

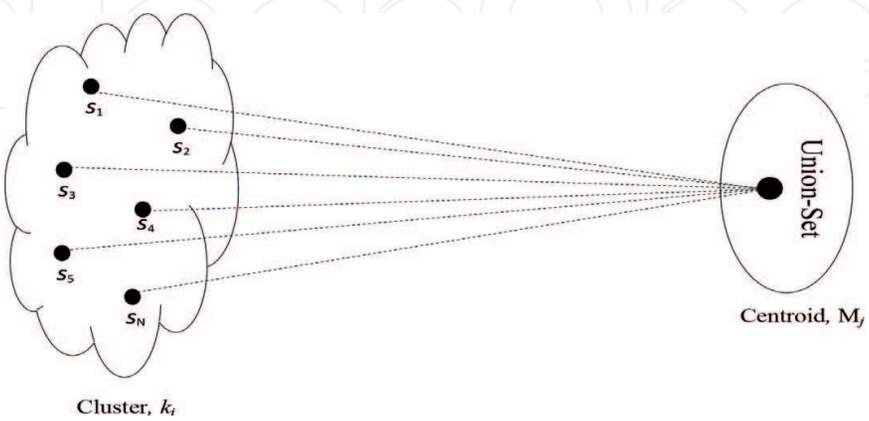

Figure 1. Clustering centroid, where $sentence_i$ (*si*) is a set of synonym words corresponding to *Sentencei* (S_i).

2.5. Calculating similarity between sentence and cluster centroid

When the CBLC algorithm calculates the semantic similarity between sentences, there are two cases to take into account. Firstly, if a sentence does belong to the cluster and secondly, if a sentence does not belong to the cluster. This case is straightforward to implement. Since the cluster centroids are represented in the same way as a union set (synonyms), the similarity between a sentence and a cluster centroid (i.e., two sentences) can be calculated by using sentence similarity measure, as described earlier. There is, however, a subtlety in the first case, which is not immediately apparent.

To demonstrate how this semantic similarity is calculated, assume that $\text{Sentence}_1 = \{word_1, word_2, word_3\}$ and $\text{Sentence}_2 = \{word_4, word_5\}$ are not semantically similar. Comparing these sentences (S_1 and S_2), we obtain the semantic vectors $\mathbf{v}_1 = \{1,1,1,0,0\}$ and $\mathbf{v}_2 = \{0,0,0,1,1\}$ which obviously have a cosine value of zero and is reliable with the fact that they are no semantic relation between them. Now suppose, however, that Sentence_1 (S_1) and Sentence_2 (S_2) are in the same cluster. If we create the cluster union set as mentioned earlier (i.e., by taking the union of all synonym words appearing in all sentences in that cluster), we obtain $M_j = \{word_1, word_2, word_3, word_4, word_5\}$. If we now calculate the semantic similarity between M_j and S_1 by using the cosine measure, we then obtain the vectors $\mathbf{v}_j = \{1,1,1,1,1\}$ and $\mathbf{v}_1 = \{1,1,1,0,0\}$, which have a similarity score equal to 0.77. An issue is clearly seen here, since S_1 and S_2 are not similar and their centroid would not carry any useful meaning. This issue in which we would not expect the similarity value like this has happened due to all of the words of S_1 already existing in the cluster centroid M_j. We can solve this problem by defining the centroid using all sentences in the cluster except the sentence with which the cluster centroid is being currently compared. Therefore, assuming that we have a cluster containing sentences $\text{Sentence}_1 \ldots \text{Sentence}_N$, and we want the similarity between this cluster and a sentence SG appearing in the cluster, we would determine the cluster centroid using only the words appearing in $Sentence_1 \cup Sentence_2 \cup \ldots \cup Sentence_{G-1} \cup_{G+1} \cup \ldots \cup Sentence_N$; that is, we omit SG in calculating the cluster centroid.

2.6. Space and time complexity of CBLC algorithm

It has been founded that the proposed algorithm is no more expansive comparing with the basic *k*-means [52] and spectral-clustering [18, 37] algorithms regarding the space complexity (i.e., the three algorithms require the storage of the same similarity scores). The time (i.e., computation) complexity of a new version of the standard *k*-means: however, far exceeds that of basic *k*-means; and spectral-clustering algorithms. Furthermore, the computation complexities appeared in the stage of calculating the similarity between each sentence and corresponding centroid; this is due to representation of the text in the sentence similarity measure we have been applied within this clustering algorithm. To demonstrate this complexity, suppose that operation time unit for calculating semantic similarity between each sentence and cluster centroid is *SentSim*, the operation time unit for recalculating cluster centroids is *ReTime*, the total number of sentences in the used dataset is *tn*, the number of clusters is *k,* and the iteration loop of the proposed algorithm is *LoopI*. Therefore, essentially, the two following computations are required for each and every clustering iteration: (i) *tn.k* times sentence to cluster centroid similarity calculation; (ii) *k* times for

relocate cluster centroid. As a result, the time complexity of proposed version can be defined as $O_{CBLC} = (SentSim.\ tn.\ k + ReTime.\ k).\ LoopI$.

Since $SentSim >> ReTime$ and $tn >> k$, the overall time complexity of CBLC algorithm is found $O(tn)$, which means that computational complexity is relative to the size of the dataset that needs to be clustered.

3. Experiments and results

This section presents the performance of the CBLC algorithm to seven benchmark datasets, and the results are compared with that of other well-known clustering algorithms; spectral clustering [18, 36], affinity propagation [35], *k*-medoids algorithm [30, 31], STC-LE [39], and *k*-means (TF-IDF) [40]. We first describe the seven benchmark datasets, discuss cluster evaluation criteria, and we then report the experimental results (**Figure 2**).

3.1. Benchmark datasets

While CBLC algorithm is obviously appropriate to tasks involving sentence clustering, the algorithm is applied to generic in nature standard datasets such as *Reuters-21,578* dataset [29], *Aural Sonar* dataset [29, 53], *Protein* dataset [29, 54], *Voting* dataset [29, 55], *SearchSnippets* [38, 56], *StackOverflow* [38], and *Biomedical* [38].

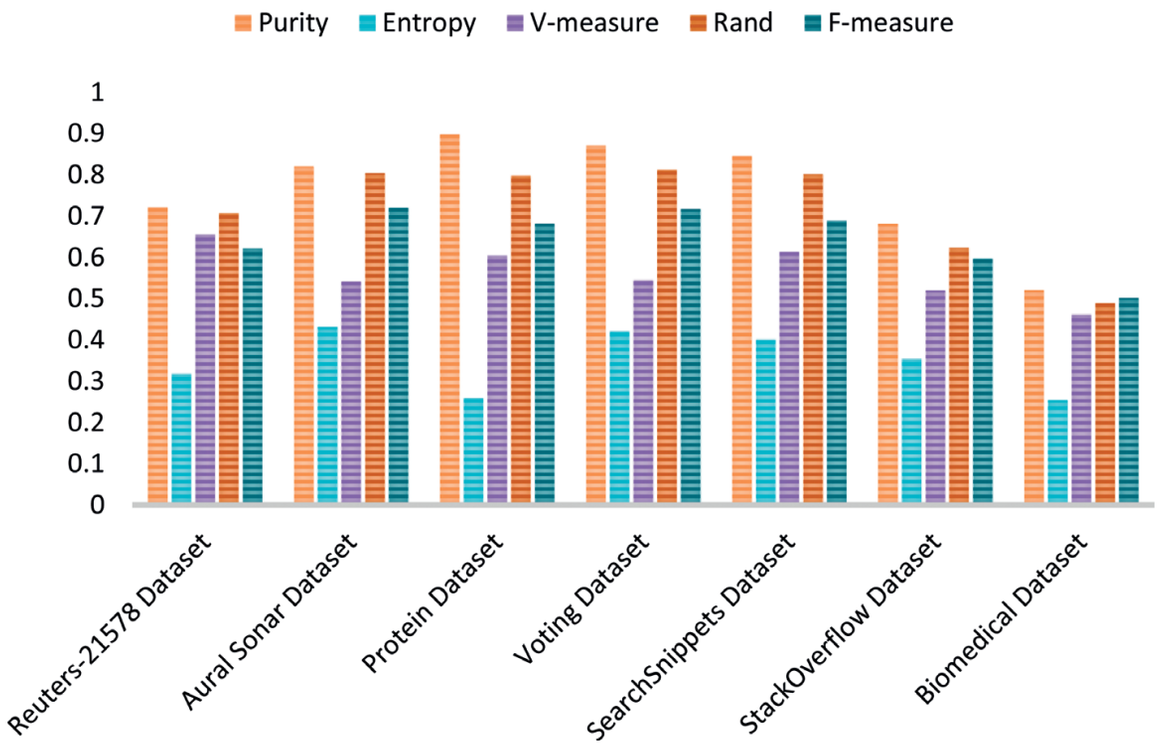

Figure 2. CBLC algorithm performance on seven benchmark datasets.

The Reuters-21,578 is the commonly used dataset for text classification task. It contains more than 20,000 documents from over 600 classes. The experimental results presented in this chapter only use a subset containing only 1833 text fragments, each of them are labeled as relating to one of 10 distinguished classes. The total number of the text fragments in each of the 10 classes is 354, 333, 258, 210, 155, 134, 113, 100, 90, and 70, respectively.

In the Aural Sonar dataset [53], two randomly selected people were asked to assign a similarity score between 1 and 5 to all pairs of signals returned from a broadband active sonar system. The two obtained scores from participated people were added to produce a 100×100 similarity matrix with values ranging from 2 to 10.

The Protein dataset [54, 57] consists of dissimilarity values for 226 samples over nine classes. We use the reduced set [57] of 213 proteins from four classes that result from removing classes with fewer than seven samples.

The Voting dataset is a two-class classification task with around 435 samples (text fragments). Similarity scores in the form of a matrix table were computed from the data in the categorical domain.

The SearchSnippets dataset consists of eight different predefined domains (i.e., classes), which was generated from the web-search-transaction result activity.

The StackOverflow dataset consists of 3,370,528 samples collected through the period of July 31, 2012, to August 14, 2012 (https//:www.kaggle.com). In this chapter, we randomly select 20,000 question titles from 20 different classes.

The Biomedical is a challenge dataset published in BioASQ's official website, and we randomly select 20,000 paper titles from 20 different MeSH major classes.

3.2. Clustering evaluation criteria

Since *complete* cluster (i.e., all objects from a single class are assigned to a single cluster) and *homogeneous* cluster (i.e., each cluster contains only objects from a single class) are hardly achieved, we aim to reach a satisfactory balance between these two approaches. Therefore, we apply five well-known clustering criteria in order to evaluate the performance of the proposed algorithm, which are *Purity, Entropy, V-measure*, *Rand Index*, and *F-measure*.

Entropy and *Purity* [58]. Entropy measure is used to show how the clusters of sentences are partitioned within each cluster, and it is known as the average of weighted values in each cluster entropy over all clusters $C = \{c_1, c_2, c_3, \ldots c_n\}$:

$$Entropy = \sum_{j=1}^{|L|} \frac{|w_j|}{N} \left(-\frac{1}{\log|C|} \sum_{i=1}^{|C|} \frac{|w_j \cap c_i|}{|w_j|} \log \frac{|w_j \cap c_i|}{|w_j|} \right) \tag{2}$$

The purity of a cluster is the fraction of the cluster size that the largest class of sentences assigned to that cluster represents, that is,

$$P_j = \frac{1}{|w_j|} \max_i \left(|w_j \cap c_i| \right) \tag{3}$$

Overall purity is the weighted sum of the individual cluster purities and is given by

$$Purity = \frac{1}{N} \sum_{j=1}^{|L|} \left(|w_j| \times P_j \right) \tag{4}$$

While purity and entropy are useful for comparing clusterings with the same number of clusters, they are not reliable when comparing clusterings with different numbers of clusters. This is because entropy and purity perform on how the sets of sentences are partitioned within each cluster, and this will lead to homogeneity case. Highest scores however, of purity and lowest scores of entropy are usually obtained when the total number of clusters is too big, where this step will lead to being lowest in the completeness. The next measure we have used considers both completeness and homogeneity approaches.

V-measure [59]. This is a measure that is known as the homogeneity and completeness harmonic mean; that is, *V* = *homogeneity* * *completeness* / (*homogeneity* + *completeness*), where *homogeneity* and *completeness* are defined as *homogeneity* = 1 – *H*(*C*|*L*)/*H*(*C*) and *completeness* = 1 – *H*(*L*|*C*)/*H*(*L*).

Eq. (5) can be written as follows, where

$$\begin{aligned} H(C) &= -\sum_{i=1}^{|C|} \frac{|c_i|}{N} \log \frac{|c_i|}{N}, \quad H(L) = -\sum_{j=1}^{|L|} \frac{|w_j|}{N} \log \frac{|w_j|}{N} \\ H(C|L) &= -\sum_{j=1}^{|L|} \sum_{i=1}^{|C|} \frac{|w_j \cap c_i|}{N} \log \frac{|w_j \cap c_i|}{|w_j|}, \quad \text{and} \quad H(L|C) = -\sum_{i=1}^{|C|} \sum_{j=1}^{|L|} \frac{|w_j \cap c_i|}{N} \log \frac{|w_j \cap c_i|}{|c_i|} \end{aligned} \tag{5}$$

Rand Index and *F-measure*. These measures depend on a combinatorial approach which considers each possible pair of sentences. It is defined as *Rand Index* = (*TP* + *FP*)/(*TP* + *FP* + *FN* + *TN*), where TP is a true positive (sentences corresponded to both same class and cluster), FP is a false positive (sentences corresponded to the different classes but same cluster), FP is a false positive (sentences corresponded to the different clusters but same class), and FN is a false negative (sentences must correspond to both different clusters and classes).

The *F*-measure is another method widely applied in the information retrieval domain and is defined as the harmonic mean of Precision (P) and Recall (R), that is, *F*-measure = 2**P***R*/(*P* + *R*), where *P* = *TP*/(*TP* + *FP*) and *R* = *TP*/(*TP* + *FN*).

3.3. Results

Since CBLC algorithm is generic in nature and can in principal be applied to any lexical semantic clustering domain, **Figure 3** shows the results of applying it to the *Reuters-21,578*, *Aural Sonar*, *Protein, Voting, SearchSnippets, StackOverflow,* and *Biomedical* datasets, respectively, by using the

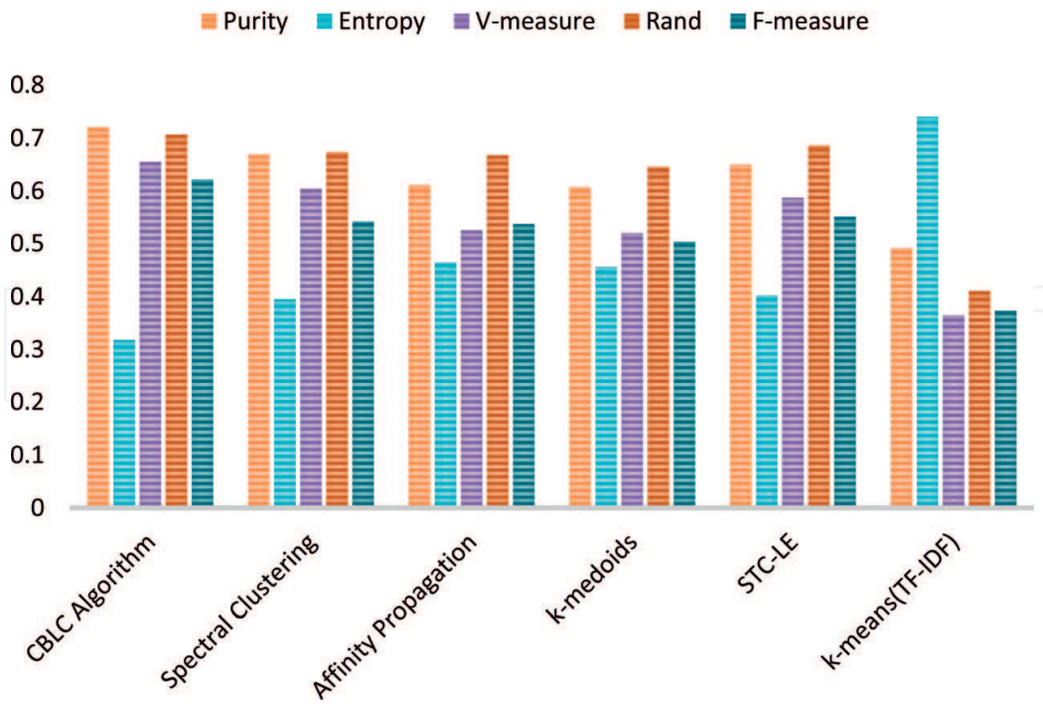

Figure 3. CBLC algorithm and other compared algorithms performance on Reuters-21,578 dataset.

Purity, Entropy, V-measure, Rand Index, and F-measure evaluation measures. CBLC algorithm however, requires an initial number of clusters in which we specified before the algorithm start. This number was varied from 7 to 12 for Reuters-21,578, Aural Sonar, Protein, Voting, and SearchSnippets datasets, and from 17 to 23 for StackOverflow and Biomedical datasets. This is because we found a proper clustering performance. Note that the values in the figure are averaged over 100 trials, and the best performance according to each measure is only presented.

Figures 3–9 show the clustering performance of CBLC algorithm comparing with that of spectral clustering, affinity propagation, *k*-medoids, STC-LE, and *k*-means (TF-IDF), respectively, on seven mentioned benchmark datasets using the five cluster evaluation criteria described earlier. For the baselined (i.e., compared) methods, the total values of the used evaluation measures (i.e., purity, entropy, V-measure, Rand Index, and F-measure) were in each measure obtained by discovering a range of numbers starting from 7 to 23 clusters and then considering that which performance is the best in overall clustering quality. The figured empirical results for our proposed new version of standard *k*-means clustering and other compared algorithms correspond to the best performance resulted from 200 time runs.

The empirical results demonstrate that CBLC algorithm significantly outperforms the other baselined algorithms on all used datasets. In this experiment however, we knew *a priori* what the real number of clusters was. Generally, we wish that the clustering algorithm could automatically determine an actual number of clusters, since we would not have this information. Even when run with a high initial number of clusters, CBLC algorithm was able to converge to a solution containing not more than seven clusters (e.g., in case of Reuters-21,578 dataset), and from the figures, it can be again seen that the evaluation of these clusterings is superior than that for the other baselined clustering algorithms.

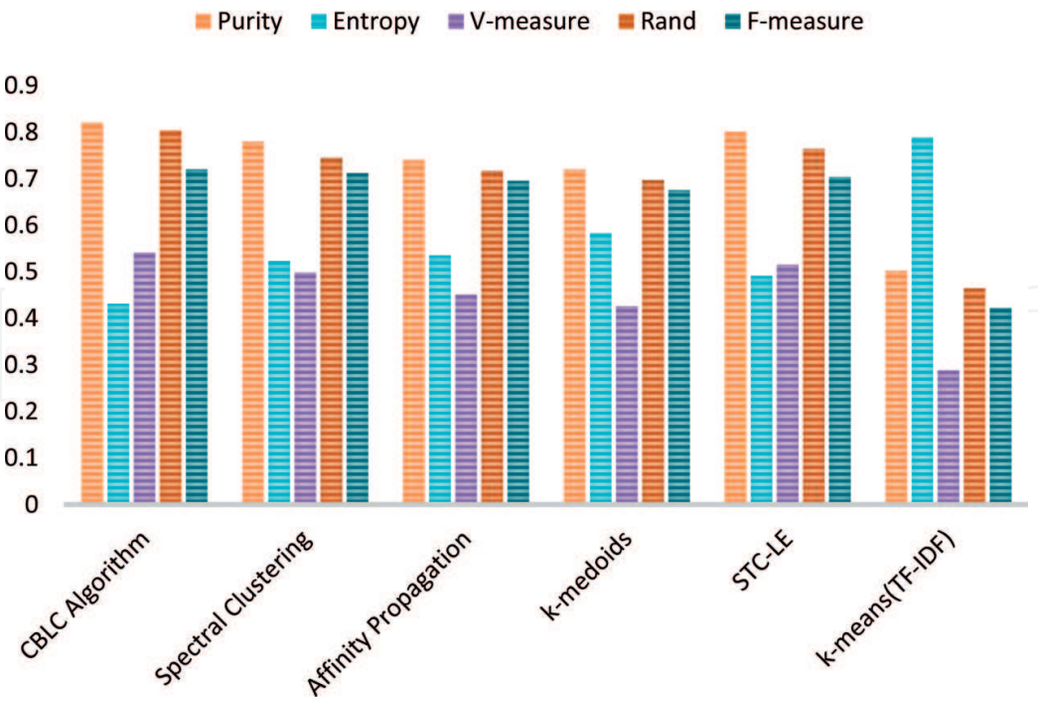

Figure 4. CBLC algorithm and other compared algorithms performance on aural sonar dataset.

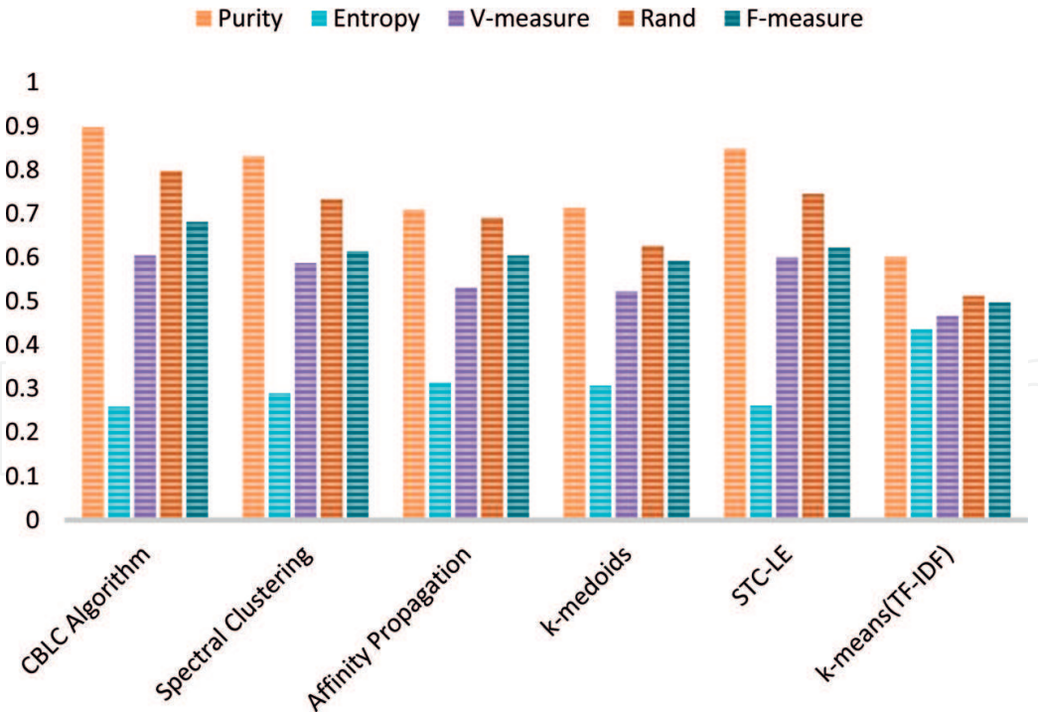

Figure 5. CBLC algorithm and other compared algorithms performance on protein dataset.

http://dx.doi.org/10.5772/intechopen.75433

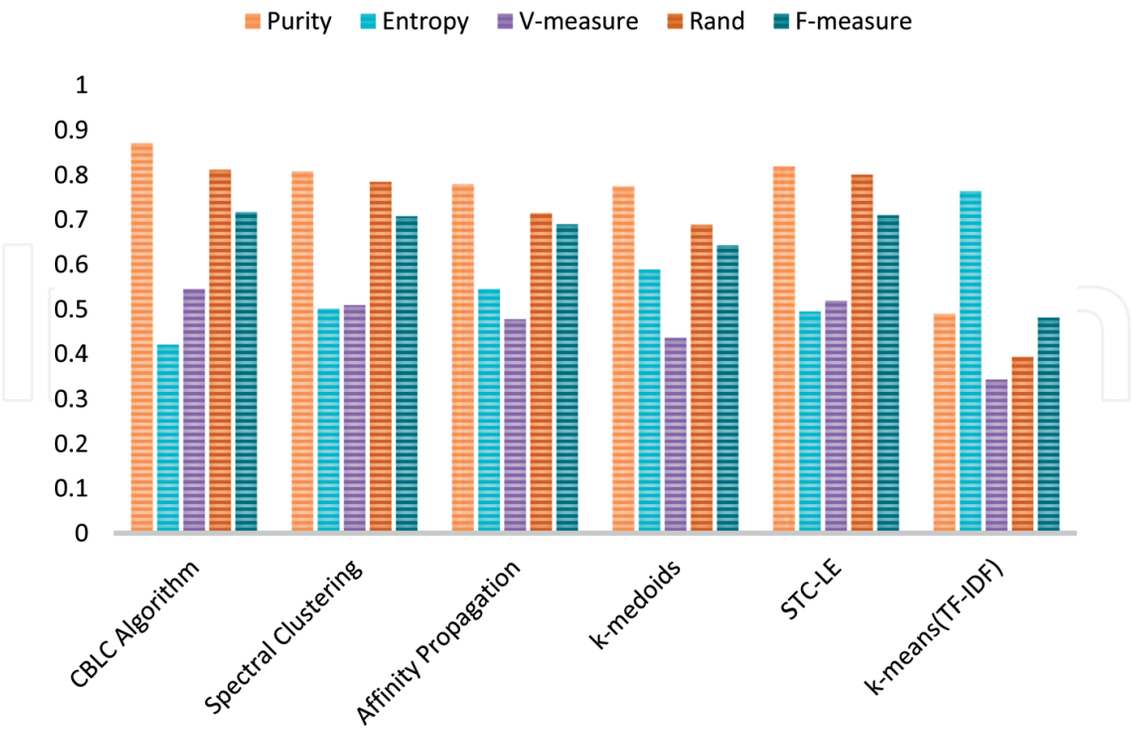

Figure 6. CBLC algorithm and other compared algorithms performance on voting dataset.

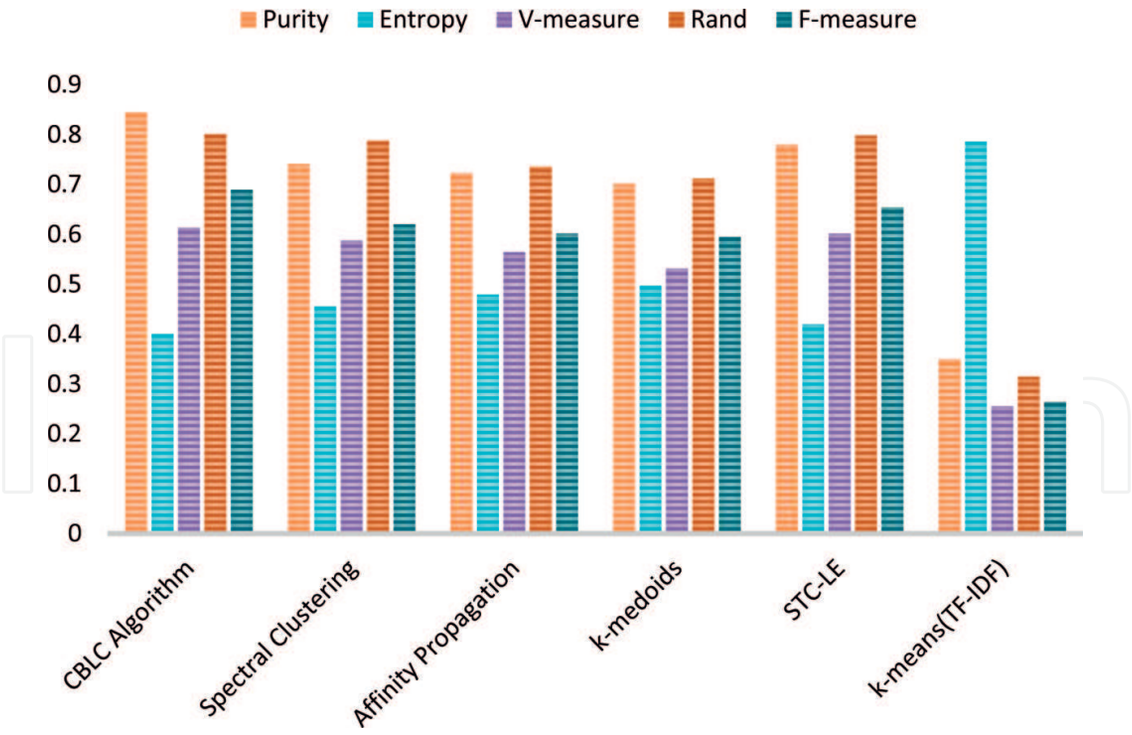

Figure 7. CBLC algorithm and other compared algorithms performance on SearchSnippets dataset.

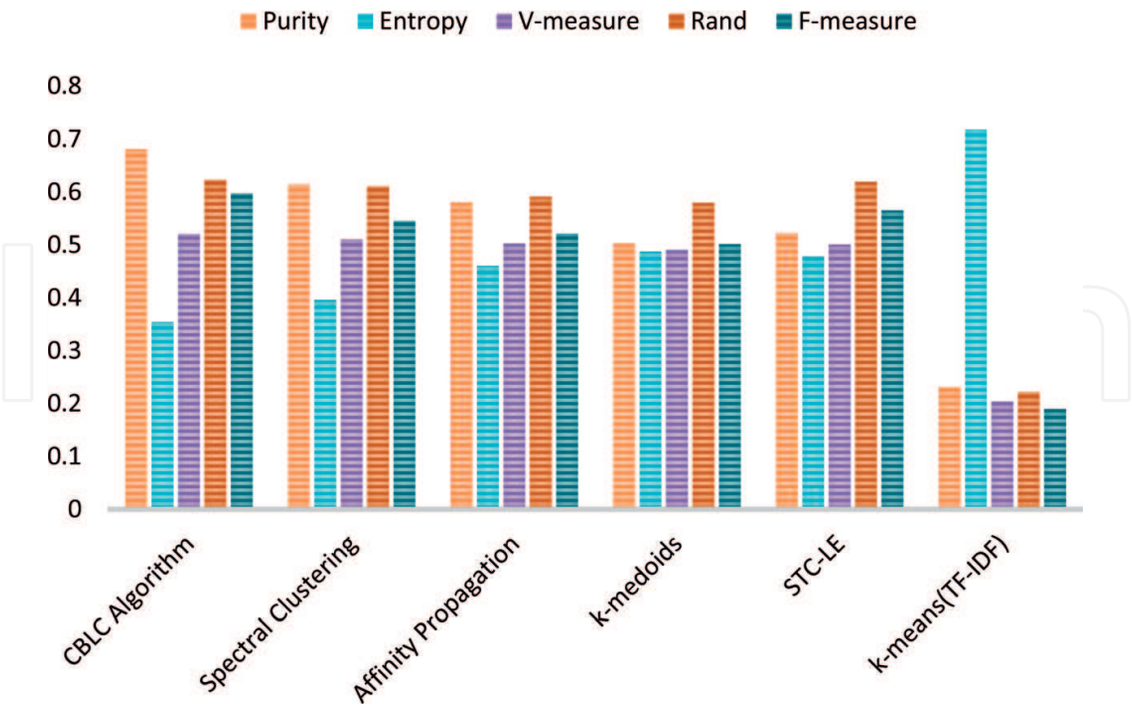

Figure 8. CBLC algorithm and other compared algorithms performance on StackOverflow dataset.

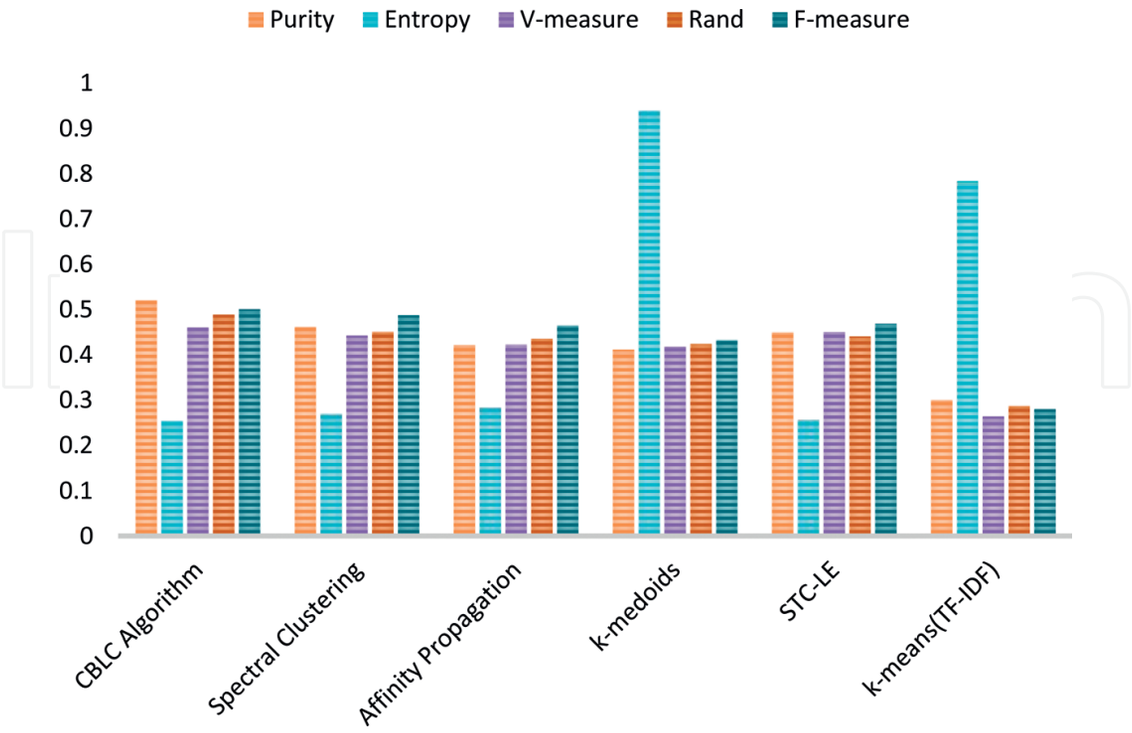

Figure 9. CBLC algorithm and other compared algorithms performance on biomedical dataset.

4. Concluding remarks

This chapter has shown a new version of the *k*-means clustering method that is able to cluster small-sized text fragments. This new variation measures the semantic similarity between patterns (i.e., sentences) based on the idea of creating a synonym expansion set to be used in the compared semantic vectors. The sentences are represented in these vectors by using semantic information derived from a WordNet that is created for the purpose of identifying the actual sense to a word, based on the surrounding context. The experimental results have demonstrated the method to achieve a satisfactory performance against the compared algorithms such as spectral clustering affinity propagation, *k*-medoids, STC-LE, and *k*-means (TF-IDF), as evaluated on several standard datasets.

A clear domain of applying the algorithm is to text-mining processing; however, the algorithm can also be used within more general text-processing settings such as text summarization. Like any clustering algorithm, the performance of CBLC will eventually be based on the text similarity values, and these values can be improved by defining the sentence-level text similarity measure that can utilize much more possible semantic information expressed with the compared sentences. Any such improvements are surly effected by the overall sentences clustering performance.

Sentence-level text clustering is an exciting area of research within the knowledge discovery and computational linguistic activities, and this chapter has proposed a new variation of *k*-means clustering which are capable to cluster sentences based on available semantic information written in these sentences. We are interested in some of the new research directions that we have experienced in this area; however, what we are most excited about is applying our proposed cluster technique to operate on the text-mining activities. This is because the concepts existing in human-written documents usually have buried knowledge and information, whereas the technique we have developed in this work is only applied on the clusters text-fragments domain. Therefore, one of the possible future works is to apply these ideas of sentence clustering to the development of complete techniques for sentiment analysis of the people's opinion.

Author details

Khaled Abdalgader

Address all correspondence to: komar@soharuni.edu.om

Sohar University, Sohar, Oman

References

[1] Li Y, McLean D, Bandar ZA, O'Shea JD, Crockett K. Sentence similarity based on semantic nets and Corpus statistics. IEEE Transactions on Knowledge and Data Engineering. 2006;**18**(8):1138-1150

[2] Mihalcea R, Corley C, Strapparava C. Corpus-based and knowledge-based measures of text semantic similarity. In: Proceedings of the 21st National Conference on Artificial Intelligence; 2006. pp. 775-780

[3] Sowmya V, Vishnu Vardhan B, Bhadri Raju MSVS. Influence of token similarity measures for semantic textual similarity. In: Proceedings of 2016 IEEE 6th International Conference on Advance Computing (IACC2016); 2016. pp. 27-28

[4] Metzler D, Dumais S, Meek C. Similarity measures for short segments of text. In: Proceedings of the 29th European Conference on Information Retrieval. 4425, Springer, Heidelberg; 2007. 16-27

[5] Islam A, Inkpen D. Semantic text similarity using corpus-based word similarity and string similarity. ACM Transactions on Knowledge Discovery from Data (TKDD). 2008;**2**(2):1-25

[6] Feng J, Zhou Y-M, Martin T. Sentence similarity based on relevance. In: Proceedings of the IPMU'08; 2008. 832-839

[7] Ramage D, Rafferty A, Manning C. Random walks for text semantic similarity. In: Proceedings of ACL-IJCNLP 2009; 2009. 23-31

[8] Achananuparp P, Hu X, Yang C. Addressing the variability of natural language expression in sentence similarity with semantic structure of the sentences. In: Proceedings of PAKDD 2009. Bangkok; 2009. 548-555

[9] Ho C, Murad MAA, Kadir RA, Doraisamy SC. Word sense disambiguation-based sentence similarity. In: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, COLING '10. Stroudsburg, PA, USA. Association for Computational Linguistics; 2010. pp. 418-426

[10] Abdalgader K, Skabar A. Short-text similarity measurement using word sense disambiguation and synonym expansion. In: Proceedings of the 23rd Australasian Joint Conference on Artificial Intelligence. (AI2010, Adelaide, Australia). vol. LNAI 6464; 2011. pp. 435-444

[11] Liu H, Wang P. Assessing sentence similarity using WordNet based word similarity. Journal of Software. June 2013;**8**(6)

[12] Zhu TT, Lan M. ECNUCS: Measuring short text semantic equivalence using multiple similarity measurements. Second Joint Conference on Lexical and Computational Semantics (SEM), Volume 1: Proceedings of the Main Conference and the Shared Task, Atlanta, Georgia, June 13-14, 2013. pp. 124-131

[13] Kenter T, Rijke DM. Short text similarity with word embeddings. In: Proceedings of the 24th ACM international conference on information and knowledge management. In CIKM '15. ACM; 2015

[14] Abdalgader K. Text-fragment similarity measurement using word sense identification. International Journal of Applied Engineering Research. 2016;**11**(24):11755-11762

[15] Abdalgader K. Computational Linguistic Techniques for Sentence-Level Text Processing". PhD Dissertation. Department of Computer Engineering and Computer Science, La Trobe University; 2011

[16] Skabar A, Abdalgader K. Improving sentence similarity measurement by incorporating sentential word importance. In: Proceedings of the 23rd Australasian joint conference on artificial intelligence. (AI2010, Adelaide, Australia). Vol LNAI 6464. 2011. pp. 466-475

[17] Abdalgader K. Word sense identification improves the measurement of short-text similarity. In: Proceedings of the International Conference on Computing Technology and Information Management (ICCTIM2014), Dubai, UAE, Digital Library of SDIWC, ISBN: 978–0–9891305-5-4. 2014. pp. 233-243

[18] Wang D, Li T, Zhu S, Ding C. Multi-document summarization via sentence-level semantic analysis and symmetric matrix factorization. In: proceedings of the 31st annual international ACM SIGIR conference on Research and Development in Information Retrieval. pp. 307-314; 2008

[19] Salton G. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley: Reading, Mass; 1989

[20] Fellbaum C, editor. "WordNet: An Electronic Lexical Database". Cambridge, MA: MIT Press; 1998

[21] Navigli R, Ponzetto S. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artificial intelligence, 193, Elsevier. 2012. pp. 217-250

[22] Abdalgader K, Skabar A. Unsupervised similarity-based word sense disambiguation using context vectors and sentential word importance. ACM Transactions on Speech and Language Processing (TSLP). 2012;**9**(2)

[23] Chen F, Han K, Chen G. An approach to sentence selection based text summarization. In: Proceedings of IEEE TENCON02; 2008. pp. 489-493

[24] Kyoomarsi F, Khosravi H, Eslami E, Dehkordy PK, Tajoddin A. Optimizing text summarization based on fuzzy logic. Seventh IEEE/ACIS International Conference on Computer and Information Science, IEEE Computer Society. 2008. pp. 347-352

[25] Radev DR, Jing H, Stys M, Tam D. Centroid-based summarization of multiple documents. Information Processing and Management: AN International Journal. 2004;**40**:919-938

[26] Aliguyev RM. A new sentence similarity measure and sentence based extractive technique for automatic text summarization. Expert Systems with Applications. 2009;**36**:7764-7772

[27] Hotho A, Nürnberger A, Paaß G. A brief survey of text mining. GLDV-Journal for Computational Linguistics and Language Technology. 2005;**20**:19-62

[28] Kosala R, Blockeel H. Web mining research: A survey. ACM SIGKDD Explorations Newsletter. 2000;**2**(1):1-15

[29] Skabar A, Abdalgader K. Clustering sentence-level text using a novel fuzzy relational clustering algorithm. IEEE Transactions on Knowledge and Data Engineering (TKDE) IEEE Computer Society Digital Library. 2013;**25**(1):62-75

[30] Kaufman L, Rousseeuw PJ. Clustering by means of medoids. In: Godge Y, editor. Statistical Analysis Based on the L_1 Norm. Amsterdam: North Holland/Elsevier; 1987. pp. 405-416

[31] Kaufman L, Rousseeuw PJ. Finding Groups in Data. Wiley; 1990

[32] Krishnapuram R, Joshi A, Liyu Y. A fuzzy relative of the k-Medoids algorithm with application to web document and snippet clustering". In: Proceedings of the IEEE Fuzzy Systems Conference; 1999. pp. 1281-1286

[33] Geweniger T, Zühlke D, Hammer B, Villmann T. Fuzzy variant of affinity propagation in comparison to median fuzzy c-means. In: Proceedings of the 7th International Workshop on Advances in Self-Organizing Maps. Springer-Verlag, Berlin, Heidelberg; 2009. pp. 72-79

[34] Geweniger T, Zühlke D, Hammer B, Villmann T. Median fuzzy C-means for clustering dissimilarity data. Neurocomputing. 2010;**73**(7–9):1109-1116

[35] Frey BJ, Dueck D. Clustering by passing messages between data points. Science. 2007;**315**: 972-976

[36] Ng AY, Jordan MI, Weiss Y. On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems. 2001:849-856

[37] Luxburg UV. A tutorial on spectral clustering. Statistics and Computing. 2007;**17**(4):395-416

[38] Xu J, Xu B, Wang P, Zheng S, Tian G, Zhao J. Self-taught convolutional neural networks for short text clustering. Neural Networks. 2017;**30**(2):117-131

[39] Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. Advances in Neural Information Processing Systems. 2001;**14**:585-591

[40] Wagsta K, Cardie C, Rogers S, Schrodl S, et al. Constrained k-means clustering with background knowledge. In: ICML. Vol. 1. 2001. pp. 577-584

[41] Ganesan KA, Zhai CX, Han J. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10). 2010

[42] Abdalgader K. Clustering short text using a centroid-based lexical clustering algorithm. IAENG International Journal of Computer Science. 2017;**44**(4):523-536

[43] Erkan G, Radev DR. LexRank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research. 2004;**22**:457-479

[44] Mihalcea R, Tarau P. TextRank: Bringing order into texts. In: Proceedings of EMNLP. 2004. pp. 404-411

[45] Brin S, Page L. The anatomy of a large-scale Hypertextual web search engine. Computer Networks and ISDN Systems. 1998;**30**:107-117

[46] Fang C, Mu D, Deng Z, Wu Z. Word-sentence co-ranking for automatic extractive text summarization. Expert Systems with Applications. 2017;**72**:189-195

[47] Namburu SM, Tu H, Luo J, Pattipati KR. Experiments on supervised learning algorithms for text categorization. IEEE Aerospace Conference. Big Sky, MT. 2005

[48] Hatzivassiloglou V, Klavans JL, Holcombe ML, Barzilay R, Kan M-Y, McKeown KR. SIMFINDER: A flexible clustering tool for summarization. In: NAACL Workshop on Automatic Summarization. Association for Computational Linguistics, 2001. 41-49

[49] Vidhya KA, Aghila GG. Text mining process, techniques and tools: An overview. International Journal of Information Technology and Knowledge Management. 2010;**2**(2):613-622

[50] Bates M. Subject access in online catalogue: A design model. Journal of the American Society for Information Science. 1986;**37**(6):357-376

[51] Jiang JJ, Conrath DW. Semantic similarity based on Corpus statistics and lexical taxonomy. In: Proceedings of the 10th International Conference on Research in Computational Linguistics. 1997. pp. 19-33

[52] MacQueen JB. Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. 1967. pp. 281-297

[53] Philips S, Pitton J, Atlas L. Perceptual feature identification for active sonar echoes. Proceedings of IEEE OCEANS Conference. 2006

[54] Hofmann T, Buhmann JM. Pairwise data clustering by deterministic annealing. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1997;**19**(1):1-14

[55] Asuncion A, Newman DJ, UCI machine learning repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science

[56] Phan X-H, Nguyen L-M, Horiguchi S. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: Proceedings of the 17th International Conference on World Wide Web, ACM. 2008. pp. 91-100

[57] Chen Y, Garcia EK, Gupta MR, Rahimi A, Cazzanti L. "Similarity-based classification: Concepts and algorithms". Journal of Machine Learning Research, vol. 10, pp. 747–776, 2009

[58] Manning CD, Raghavan P, Schütze H. Introduction to Information Retrieval. Cambridge: Cambridge University Press; 2008

[59] Rosenberg A, Hirschberg J. V-Measure: A conditional entropy-based external cluster evaluation measure. In: Proceedings of the EMNLP. 2007. pp. 410-420

IntechOpen