View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 82 (2016) 28 – 34

Procedia Computer Science

Symposium on Data Mining Applications, SDMA2016, 30 March 2016, Riyadh, Saudi Arabia

Finding similar documents using different clustering techniques

Sumayia Al-Anazi, Hind AlMahmoud, Isra Al-Turaiki*

Information Technology Department
College of Computer and Information Sciences
King Saud University
Riyadh 12372, Saudi Arabia

Abstract

Text clustering is an important application of data mining. It is concerned with grouping similar text documents together. In this paper, several models are built to cluster capstone project documents using three clustering techniques: *k-means*, *k-means fast*, and *k-medoids*. Our datatset is obtained from the library of the College of Computer and Information Sciences, King Saud University, Riyadh. Three similarity measure are tested: *cosine similarity*, *Jaccard similarity*, and *Correlation Coefficient*. The quality of the obtained models is evaluated and compared. The results indicate that the best performance is achieved using *k- means* and *k-medoids* combined with cosine similarity. We observe variation in the quality of clustering based on the evaluation measure used. In addition, as the value of k increases, the quality of the resulting cluster improves. Finally, we reveal the categories of graduation projects offered in the Information Technology department for female students.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of the Organizing Committee of SDMA2016

Keywords: clustering; k-means; k-medoids; text mining; data mining; cosine similarity

1. Introduction

Today, with the rapid advancements in technology we are able to accumulate huge amounts of data of different kinds. *Data mining* emerged as a field concerned with the extraction of useful knowledge from data[1]. Data mining techniques have been applied to solve a wide range of real-world problems. Clustering is an unsupervised data mining technique where the labels of data objects are unknown. It is the job of the clustering technique to identify the categorisation of data objects under examination. Clustering can be applied to different kinds of data including text. When dealing with textual data, objects can be documents, paragraphs, or words[2]. *Text clustering* refers to the process of grouping similar text documents together. The problem can be formulated as follows: given a set of documents it is required to divide them into multiple groups, such that documents in the same group are more similar to each other than to documents in other groups. There are many applications of text clustering including: document organisation and browsing, corpus summarisation, and document classification[3].

* Corresponding author. Tel.: +0-966-11805-2909.
E-mail address: sumaiah.j@gmail.com, hindsalmahmoud@gmail.com, ialturaiki@ksu.edu.sa

1877-0509 © 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Organizing Committee of SDMA2016
doi:10.1016/j.procs.2016.04.005

Traditional clustering techniques can be extended to deal with textual data. However, there are many challenges in clustering textual data. The text is usually represented in high dimensional space even when it is actually small. Moreover, correlation between words appearing in the text needs to be considered in the clustering task. The variations in document sizes is another challenge that affects the representation. Thus, the normalisation of text representation is required[2].

In this paper, we use data mining techniques in order to cluster capstone projects in information technology. In particular, we study graduation projects offered in the Information Technology department (IT) for female students at the College of Computer and Information Sciences, King Saud University, Riyadh. The goal is to to reveal the areas that the department encourages students to work on. The results of the study will be beneficial to both students and decision makers. For students, clustering graduation projects will help them find previous projects related to their own project idea. The study will also help the administration make right decisions when approving project proposals. We apply and compare three clustering techniques: *k-means*[4], *k-means fast*[5], and *k-mediods*[6]. In addition, three similarity measures are used to form clusters: *cosine similarity*[7], *Jaccard similarity*, and *Correlation Coefficient*[1]. The goal of the comparison is to find the best combination of clustering technique and similarity measure and to study the effect of increasing the number of clusters, k.

The rest of the paper is organised as follows: In Section 2, we review some of the literature in the field of text clustering. Section 3, describes our dataset, the steps taken to prepare it for data mining, and the data mining techniques and the similarity measures used in our experiment. The cluster evaluation measures and our main findings are discussed in Section 4. Finally, our paper concludes in Section 5.

2. Literature Review

Text clustering is one of the important applications of data mining. In this section, we review some of the related work in this field. Luo et al.[3] used the concepts of document *neighbors* and *links* in order to enhance the performance of *k-means* and *bisecting k-means* clustering. Using a pairwise similarity function and a given similarity threshold, the neighbors of a document are the documents that are considered similar to it. A *link* between two documents is the number of common neighbors. The concepts were used in the selection of initial cluster centroids and in document similarity measuring. Experimental results using 13 datasets showed better performances as compared to the standard algorithms.

Bide and Shedge[8] proposed a clustering pipeline to improve the performance of *k-means* clustering. The authors adopted a divide-and-conquer approach to cluster documents in the *20 Newsgroup dataset*[9]. Documents were divided into groups where preprocessing, feature extraction, and *k-means* clustering were applied on each group. Document similarity was calculated using the cosine similarity measure. The proposed approach achieved better results as compared to standard *k-means* in terms of both cluster quality and execution time.

Mishra et al.[10] used *k-means* technique to cluster documents based on themes present in each one. The main assumption was that a document may deal with multiple topics. The proposed approach, called *inter-passage based clustering*, was applied to cluster document segments based on similarity. After segments were preprocessed, keywords were identified for each segment using *term frequency-inverse document frequency*[11] and sentiment polarity scores[12]. Each segment was then represented using keywords and a segment score was calculated. Finally, *k-means* was applied to all segments. The resulting clusters showed high intra-cluster similarity and low inter-cluster similarity.

In general, algorithms used for clustering texts can be divided into: agglomerative, partitioning-based, and probabilistic-based algorithms[13]. Agglomerative algorithms iteratively merge documents into clusters based on pairwise similarity. The resulting clusters are organised into a cluster hierarchy (also *dendogram*). In partitioning algorithms, documents are split into mutually exclusive (non-hierarchical) clusters. The splitting process optimises the distance between documents within a cluster. Probabilistic clustering is based on building generative models for the documents. Partitioning algorithms for text clustering have been extensively studied in the literature. This is mainly due to the low computational requirements as compared to other clustering algorithms. In this paper, we choose to utilize three partitioning-based algorithms: *k-means*[4], *k-means fast*[5], and *k-mediods*[6] in order to cluster capstone projects.

3. Methodology

3.1. Data collection and preprocessing

The dataset was collected manually from the library of the College of Computer and Information Sciences, King Saud University, Riyadh. We selected capstone projects with dates between 2010 to 2014. A total of 63 projects were collected. For each project, the following attributes were considered: project title, abstract, and supervisor name. Pre-processing was conducted as follows:

1. Tokenization: the first step was to split text into element called *tokens*[14]. A token can be a symbol, a word, a phrase, or a sentence. We split our datatset into tokens using whitespace as a delimiter.
2. Filtering: The result of the tokenisation step was filtered to remove meaningless words. Filtering was done based on minimum length. All tokens with lengths less than three characters were removed.
3. Stemming: this is an important step in text mining where words are reduced to their root forms.
4. Cases Transformation: finally, all the words were converted to lowercase.

3.2. Document Representation

The *vector space model*[7] is a common representation of text documents. Let D be a collection of documents and let $T = \{t_1, t_2, ..., t_n\}$ be the set of terms appearing in D. A document $x \in D$ can be represented as an $n-dimensional$ vector in the term space T. Let w_{x,t_i} be the number of times a term $t_i \in T$ appears in x, then the vector of x is defined as:

$$x = \{w_{x,t_1}, w_{x,t_2}, ..., w_{x,t_n}\} \tag{1}$$

3.3. Data Mining

Clustering Algorithms: *k-means* and *k-medoids* are well-known and widely applicable clustering algorithms. Here, we provide a brief description of these algorithms.

- *k-means*[4] is an iterative clustering algorithm. It is based on partitioning data points into *k* clusters using the concept of *centroid*. The cluster *centroid* is the mean value of the data points within a cluster. The produced partitions feature high intra-cluster similarity and inter-cluster variation. The number of clusters, *k*, is a pre-determined parameter of the algorithm. *k-means* works as follows: 1) *k* data points are arbitrarily selected as cluster centroids. 2) the similarity of each data point to each cluster centroid is calculated. Then data point are re-assigned to the cluster of the closest centroid. 3) the k centroids are updated based on the newly assigned data points. 4) steps 2 and 3 are repeated until convergence is reached.
- *k-medoid*[6] is a partitioning-based clustering algorithm similar to *k-means*. However, the *k-medoid* algorithm uses actual data points to represent clusters. The algorithm is less sensitive to outliers than *k-means* and works as follows: 1) *k* data points are arbitrarily selected to form the set of current cluster representatives (medoids). 2) the remaining data points are assigned to the cluster of the closest representative. 3) a data point that is not in the current set of cluster representatives is randomly selected. 4) the total cost of replacing one of the cluster representative points with the randomly selected one is calculated. 4) the replacement takes place only if the quality of the resulting clusters is improved. 5) steps 2-4 are repeated until no improvement can be achieved.
- *k-means fast*[5] is an accelerated version of *k-means* where many un-necessary distance calculations are avoided using triangle inequality. The *k-means fast* algorithm is suitable for larger values of *k* and for datasets with large number of attributes. However, it requires more memory.

Similarity Measures: There are many metrics for measuring document similarity. We focus on three common measures in this domain which are: *cosine similarity*[7], *Jaccard similarity coefficient*, and *Correlation Coefficient*.

Cosine similarity measures the cosine of the angle between the vectors of two documents. Given two vectors x and y, each of length n, the cosine similarity can be calculated as follows:

$$cos(x, y) = \frac{x.y}{||x||||y||} \tag{2}$$

where $||x|| = \sqrt{x_1^2 + x_2^2 + ... + x_n^2}$. The *Jaccard similarity coefficient*, also known as *Jaccard index*, is a popular measure of similarity and is calculated as follows:

$$Jaccard(x, y) = \frac{q}{q + r + s} \tag{3}$$

where, q is the total number of terms that are present in both documents, r is total number of terms that are present in x but not in y, and s is the total number of terms that are present in y but not in x. The value of both cosine and Jaccard range between 0 (no similarity) and 1 (identical matches).

The *correlation coefficient* can be used to measure the degree of relatedness for two vectors. The value of *correlation coefficient* ranges from -1 (negative correlation) and 1 (positive correlation). The *correlation coefficient* can be calculated as follows:

$$r(x, y) = \frac{n \sum_{t=1}^{n} w(x,t).w(y,t) - TF_x.TF_y}{\sqrt{[n \sum_{t=1}^{n} w(x,t)^2 - TF_x^2].[n \sum_{t=1}^{n} w(y,t)^2 - TF_y^2]}} \tag{4}$$

where, $TF_x = \sum_{t=1}^{n} w(x,t)$

4. Experimental Results

Here, we cluster our dataset using *k-means*[4], *k-means fast*[5], and *k-mediods*[6]. With each clustering technique, we build models using different values of k and the three similarity measures described above. The RapidMiner[15] platform was used in our experiment. This open source platform provides a friendly GUI and supports all the steps of *Knowledge Discovery from Data*, including: data pre-processing, data mining, model validation, and result visualisation. Figure 1 shows the main steps of this study.

4.1. Evaluation Measures

We evaluated and compared the quality of the obtained clustering models using two cluster evaluation measures: the *average within cluster distance*[15] and *Davies-Bouldin Index* (DB)[16]. The *average within cluster distance* is defined as the average of the distance between a cluster centroid and all elements in a cluster. As for DB, given as set of clusters, this metric measures the average similarity between each cluster and its most similar one. This metric is calculated as follows:

$$DB = \frac{1}{n} \sum_{i=1, i \neq j}^{n} max \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j))} \right) \tag{5}$$

where n is the number of clusters, σ_i is the average distance of all elements in cluster i to the cluster centroid c_i, and σ_j is the average distance of all elements in cluster j to the cluster centroid c_j, and $d(c_i, c_j)$ is the distance between the clusters centroids c_i and and c_j. Since the optimal clusters should be compact and have the least similarity to each other, the value of DB should be minimised.

4.2. Result Discussion

In this section, we discuss the quality of the obtained clustering models based on the values of the clustering evaluation measures. We compare all the obtained models to find the best combination of clustering technique and similarity measure. In addition, we look into individual clustering algorithms to find for each, the best similarity measure. Tables 1, 2, and 3 summarise our experimental results.

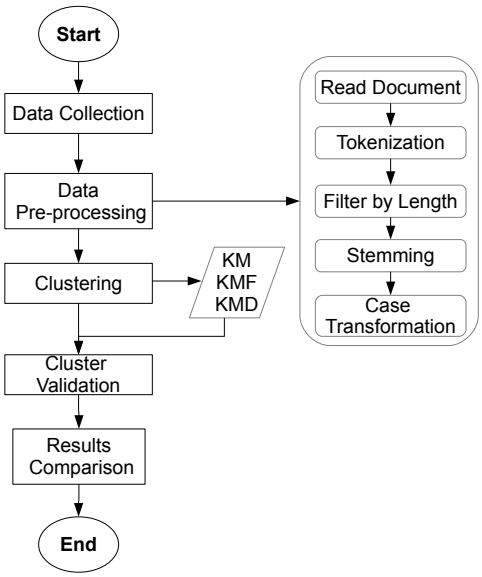

Fig. 1. The steps performed in our experiment

Table 1. The accuracy of the clustering techniques using cosine similarity

	Average within cluster distance			Davis Bouldin Index		
k	KM	KMF	KMD	KM	KMF	KMD
5	0.872	0.872	1.696	4.623	4.623	1.788
6	0.852	0.852	1.636	4.082	4.082	1.759
7	0.834	0.834	1.602	3.962	3.962	1.683
8	0.816	0.816	1.571	3.685	3.685	1.711
9	0.792	0.792	1.511	3.316	3.316	1.655
10	0.773	0.773	1.457	3.237	3.237	1.596

Based on the average within cluster distance, the results indicate that *k-means* and *k-means fast* perform similarly when the *cosine similarity* is used. This could be partially due to the ability of the cosine similarity measure to ignore document length. However, *k-means* outperforms both *k-means fast* and *k-medoids* for the *Jaccard similarity* and *correlation coefficient*. In terms of DB index, *k-medoids* shows better performance than *k-means* and *k-means fast* for all similarity measures. The worst performance is obtained with *k-means fast* and *Jaccard similarity*. For all clustering techniques, the best average within cluster distance is achieved when the *cosine similarity* is used.

We observed variation in the quality of clustering of *k-means* and *k-means fast*. The two clustering techniques show better quality when the *average within cluster distance* is used. As for *k-medoids*, the quality of clustering is similar regardless of the evaluation measure used. We found that the quality of clustering models improves as the value of k increases. Overall, the best performance is obtained using *k-means* and *k-medoids* combined with *cosine similarity*.

As shown in Figure 2, we found that capstone project ideas can be generally divided into the following categories: E-health applications, Arabic and Islamic applications, location-based applications, voice, image, and signal recognition, games, and e-learning applications.

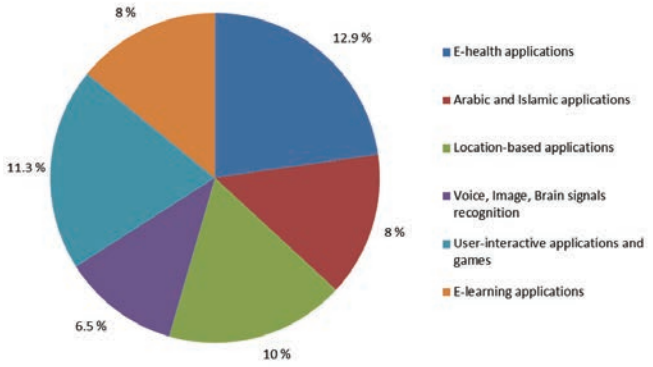

Fig. 2. The clusters obtained in our experiment

Table 2. The accuracy of the clustering techniques using Jaccard similarity

	Average within cluster distance			Davis Bouldin Index		
k	KM	KMF	KMD	KM	KMF	KMD
5	0.882	0.897	1.697	4.719	∞	1.810
6	0.862	0.890	1.665	4.365	∞	1.790
7	0.835	0.876	1.631	3.815	∞	1.745
8	0.819	0.958	1.597	3.650	∞	1.718
9	0.802	0.896	1.563	3.392	∞	1.675
10	0.786	0.895	1.527	3.212	∞	1.666

Table 3. The accuracy of the clustering techniques using correlation coefficient

	Average within cluster distance			Davis Bouldin Index		
k	KM	KMF	KMD	KM	KMF	KMD
5	0.882	0.884	1.720	4.691	4.414	1.817
6	0.864	0.868	1.667	4.367	4.161	1.783
7	0.840	0.855	1.639	3.905	4.113	1.747
8	0.836	0.857	1.608	∞	∞	∞
9	0.804	0.848	1.569	3.457	∞	1.680
10	0.789	0.846	1.538	3.330	∞	1.609

5. Conclusion

We built several clustering models for graduation project documents at King Saud University. Three cluster similarity measures were tested and the quality of the resulting clusters was evaluated and compared. We found that the best performance can be obtained using *k- means* and *k-medoids* combined with *cosine similarity*. The documents in our dataset were of various lengths and fell into different topics. Since the cosine similarity measure is independent of document length, it was able to better deal with our dataset. There was a variation in the quality of clustering based on the cluster evaluation measure used. We also found that as the value of k increased, the quality of the resulting clusters improved. Finally, we concluded that project ideas usually fall into the following categories: E-health applications, Arabic and Islamic applications, location-based applications, voice, image, and signal recognition, games, and e-learning applications. As a future work, we plan to build a system using these clustering techniques to help students find similar projects. The system should also serve as a repository of capstone project documents, since no similar system exists.

References

1. Han, J., Kamber, M.. *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann; 3rd ed.; 2011. ISBN 978-0-12-381479-1.
2. Aggarwal, C.C., Zhai, C.. A Survey of Text Clustering Algorithms. In: Aggarwal, C.C., Zhai, C., editors. *Mining Text Data*. Springer US; 2012, p. 77–128.
3. Luo, C., Li, Y., Chung, S.M.. Text document clustering based on neighbors. *Data & Knowledge Engineering* 2009;**68**(11):1271–1288.
4. Hartigan, J.A.. *Clustering Algorithms*. New York, NY, USA: John Wiley & Sons, Inc.; 99th ed.; 1975. ISBN 978-0-471-35645-5.
5. Elkan, C.. Using the Triangle Inequality to Accelerate k-Means. In: Fawcett, T., Mishra, N., editors. *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*. AAAI Press; 2003, p. 147–153.
6. Kaufman, L. and Rousseeuw, P.J., . Clustering by means of Medoids. In: Y. Dodge and North-Holland, , editor. *Statistical Data Analysis Based on the L1-Norm and Related Methods*. Springer US; 1987, p. 405–416.
7. Blair, D.C.. Information Retrieval, 2nd ed. C.J. Van Rijsbergen. London: Butterworths; 1979: 208 pp. Price: $32.50. *Journal of the American Society for Information Science* 1979;**30**(6):374–375.
8. Bide, P., Shedge, R.. Improved Document Clustering using k-means algorithm. In: *2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*. 2015, p. 1–5.
9. Lang, K.. 20 newsgroups data set. 2008 (accessed 2015-12-18). "http://www.ai.mit.edu/people/jrennie/20Newsgroups/".
10. Mishra, R., Saini, K., Bagri, S.. Text document clustering on the basis of inter passage approach by using K-means. In: *2015 International Conference on Computing, Communication Automation (ICCCA)*. 2015, p. 110–113.
11. Salton, G., Buckley, C.. Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 1988;**24**(5):513–523.
12. Esuli, A., Sebastiani, F.. SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining. In: *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06*. 2006, p. 417–422.
13. Aggarwal, C.C., Zhai, C.. *Mining Text Data*. Springer Science & Business Media; 2012. ISBN 978-1-4614-3223-4.
14. Verma, T., Renu, , Gaur, D.. Tokenization and filtering process in rapidminer. *International Journal of Applied Information Systems* 2014; **7**(2):16–18.
15. Home - RapidMiner Documentation. 2015 (accessed 2015-12-18). "http://docs.rapidminer.com/".
16. Davies, D.L., Bouldin, D.W.. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1979; **PAMI-1**(2):224–227.