Automatic assessment of online discussions using text mining

Authors:
Yvette Awuor[1]
Robert Oboko[2]

Affiliations:
[1]Department of Computer Science and Business Information Technology, Kenya Methodist University, Kenya

[2] School of Computing and Informatics, University of Nairobi, Kenya

Correspondence to:
Robert Oboko

Email:
roboko@uonbi.ac.ke

Postal address:
University of Nairobi, School of Computing and Informatics, PO Box 30197-00100, Nairobi

Dates:
Received: 19 Apr. 2011
Accepted: 26 Oct. 2011
Published: 29 May 2012

How to cite this article:
Awuor Y, Oboko R. Automatic assessment of online discussions using text mining. Int J Machine Learn Appl. 2012;1(1), Art. #2, 7 pages. http://dx.doi.org/10.4102/ijmla.v1i1.2

Online discussion forums have rapidly gained usage in e-learning systems. This has placed a heavy burden on course instructors in terms of moderating student discussions. Previous methods of assessing student participation in online discussions followed strictly quantitative approaches that did not necessarily capture the students' effort. Along with this growth in usage there is a need for accelerated knowledge extraction tools for analysing and presenting online messages in a useful and meaningful manner. This article discussed a qualitative approach which involves content analysis of the discussions and generation of clustered keywords which can be used to identify topics of discussion. The authors applied a new k-means++ clustering algorithm with latent semantic analysis to assess the topics expressed by students in online discussion forums. The proposed algorithm was then compared with the standard k-means++ algorithm. Using the Moodle course management forum to validate the proposed algorithm, the authors show that the k-mean++ clustering algorithm with latent semantic analysis performs better than a stand-alone k-means++.

Introduction

Technology is increasingly being embraced as an efficient tool which provides increased flexibility for learners in higher education. Computer-mediated communication is a key element of e-learning systems and strategies. Online discussions are one of the most important applications of computer-mediated communication in e-learning environments. This is because they provide an asynchronous collaborative learning environment where interaction takes place between group members, and they have been included in many learning management systems.

Online discussion boards are a promising strategy for promoting collaborative problem-solving courses and discovery-oriented activities. Online discussions offer a number of potential benefits that can help engage students in activities that contribute to their intellectual growth. For example, composing a response in online discussions often requires greater reflection than in face-to-face discussions. Other benefits include promotion of team-building and critical thinking and support for collaborative work.

Most systems for assessing online discussion are based on quantitative approaches.[1] A common method for assessing a student's contribution is to count the number of postings. As course enrolments increase, heavier online interaction can place a considerable information load on course instructors. Discussion postings are sometimes very short, many consisting of only one or two words, or not relevant to the problem under discussion. Thus students do not fully exploit this collaborative problem-solving environment in which they could discuss relevant technical issues with one another.[2] There is a need to encourage students to participate in online discussions by monitoring the content of their messages. This would encourage them to put more effort into their studies.

According to Song and Park[3], text mining (also referred to as text data mining) can be defined as a knowledge-intensive process in which a user interacts with a collection of postings over time using a suite of analysis tools. In a manner analogous to data mining, text mining seeks to extract useful information from data sources through identification and exploration of interesting patterns. The purpose of text mining in unstructured textual information is to extract meaning numerical indices, and hence make the information in the text accessible to data-mining techniques. Data mining is the extraction of hidden predictive information from large databases using statistical methods and machine learning algorithms. In the case of text mining, however, the data sources are natural language texts, and interesting patterns are found not amongst formalised database records but in the unstructured textual data. Both techniques aim to find hidden patterns and relationships in data.

In data mining the information is implicit in the data; it is hidden, unknown and could hardly be extracted without automatic data-mining techniques. However, with text mining the information

to be extracted is explicitly expressed in the text. Text mining aims to bring out such information in a manner that is amenable for processing by computers directly, with no need for a human intermediary.

As course management systems gain popularity in facilitating teaching, a discussion forum is a key component to facilitate interactions amongst students and teachers. Content analysis is the most popular way to study a discussion forum. However, content analysis is both labour-intensive and time- and energy-consuming. In an asynchronous virtual learning environment an instructor needs to keep monitoring the discussion forum in order to maintain the quality of the forum; hence the application of text-mining techniques in online discussions to enable instructors to assess the content of discussions easily by identifying discussion topics.

Discussion boards in e-learning environments contain a wealth of knowledge that is frequently stored in a database and rarely used beyond the initial posting and response situation.[4] Organising and extracting this information can provide a tool to assess learning in discussion boards. Moreover, students' online discussion forums can generate long, unstructured responses that can result in a greater information load for the instructor to read and assess. Assessing students' contributions hence becomes difficult and time-consuming for the course instructors. Most methods of assessing students' participation in online discussions follow a strictly quantitative approach that does not necessarily correlate with learning or students' effort.[2] Indeed this may encourage students simply to post frequent messages, without making a serious attempt to address the problem under discussion.

Mining and extracting quality knowledge from online discussions is thus significant for assessing the participation of students. This article proposes the use of a text-mining technique to analyse the content of student discussions. Specifically the study employed k-means++ clustering using an optimised latent semantic analysis algorithm to capture semantic concepts hidden in words in a discussion posting. The algorithm is used to discover the associated patterns between words and their corresponding concepts in discussions.

The k-means++ algorithm is used for choosing initial values for clustering, with the aim of spreading the k initial centres away from each other. Latent semantic analysis is an automatic method that requires a singular value decomposition (SVD) technique to decompose a large term-by-document matrix into a set of k orthogonal factors. The reduced space hopefully captures the true relationships between documents.[3]

The rest of this article is organised as follows: firstly a review of related work and literature on text mining and the k-means++ clustering algorithm, followed by presentation of our proposed approach and experimental set-up, results, discussion and conclusions.

Review of related work

Song and Park[3] proposed latent semantic analysis-based k-means clustering with the aim of improving scalability and lowering computation costs. They used latent semantic analysis to reduce the large vector space model into a reduced latent semantic space using SVD.

They validated the effectiveness of their algorithm by comparing it with k-means applied in a vector space model. Their experimental results demonstrated that k-means applied in latent semantic analysis is more superior to conventional k-means used in vector space models. The analyses of latent semantics showed that a latent semantic analysis model not only provided an underlying semantic structure but also drastically reduced dimensionality, which is very suitable for clustering algorithms.[3]

In another approach Paulsen and Ramampiaro[5] investigated the effects of combining latent semantic indexing and a flat clustering method for retrieval of biomedical information. They proposed a two-step k-means algorithm as an improvement to the standard k-means algorithm, which they considered too greedy since it builds up a solution by gobbling up the choice that offers the most obvious and immediate benefit. They created initial clusters based on latent semantic indexing. The centroids were calculated based on the distributed documents, such that their values were higher than a threshold value of 0.8. Documents were compared to the centroids and a document assigned to a cluster only if the similarity value for the document and the cluster centroid exceeded the threshold value. The main aim was to force centroids away from each other, thereby making the algorithm less greedy. Their results showed that their two-step algorithm performed better than standard k-means.

Li and Wu[6] studied online forums' hotspot detection and forecasting using sentiment analysis and text-mining approaches. First they created an algorithm to automatically analyse the emotional polarity of a text and obtain a value for each piece of text. Then they combined the algorithm with k-means clustering and a support vector machine to develop an unsupervised text-mining approach. They used the proposed text-mining approach to group the forums into various clusters, the centre of each representing a hotspot forum within the current time span. The data sets used in their empirical studies were acquired and formatted from Sina sports forums. Experimental results demonstrated that support vector machine forecasting achieves results highly consistent with k-means clustering. They further proposed a model that would detect topics in discussion forums. Stavrianou, Chauchat and Velcin[7] proposed a new framework for discussion analysis based on message-based graphs where each vertex represented a message object and each edge pointed out which message the specific node replied to. The edges were weighted by keywords that characterised the exchanged messages. The model allowed a content-oriented representation of the discussion and facilitated identification of discussion chains. They subsequently compared the two

representations (user-based and message-based graphs) and analysed the different information that can be extracted from them. Their experiments with real data validated the proposed framework and showed the additional information that can be extracted from a message-based graph. The study by Stavrianou et al.[7] relates to the present study since it proposed a framework that modelled online discussions. The model enabled content-oriented representation and allowed identification of the interesting discussion parts. It also facilitated classification of the discussion from the point of view of topics discussed.

Text mining

Text mining is used to denote any system that analyses large quantities of natural language text and detects lexical or linguistic usage patterns in an attempt to extract useful information.[8] It may be loosely characterised as the process of analysing text to extract information that is useful for specific purposes. Compared with the kind of data stored in databases, text is unstructured, amorphous, and difficult to deal with algorithmically. Nevertheless, in conventional communication text is the most common vehicle for formal exchange of information. The field of text mining usually deals with texts whose function is communication of factual information or opinions, and the motivation for trying to extract information from such text automatically is compelling.[3]

Text-mining algorithms operate on feature-based representations of documents. Different features can be employed to represent documents, as indicated below:

- characters
- words
- terms
- concepts.

Characters

The individual component-level letters, numerals, special characters and spaces are the building blocks of higher-level semantic features such as words, terms and concepts. A character-level representation can include the full set of all characters for a document or some filtered subset.

Words

Specific words selected directly from a document are what might be described as the basic level of semantic richness. For this reason word-level features are sometimes referred to as existing in the original feature space of a document. In general, a single word-level feature should equate with or have the value of no more than one linguistic token.

Terms

These are single words and multiword phrases selected directly from the corpus of a document by means of term-extraction methodologies. Term-level features can only be made up of specific words and expressions found within the document for which they are meant to be generally representative. Hence, a term-based representation of a document is necessarily composed of a subset of the terms in that document.

Concepts

These are features generated for a document by means of manual, statistical, rule-based, or hybrid categorisation methodologies. Concept-level features can be manually generated for documents but are more commonly extracted from documents using complex pre-processing routines that identify single words, multiword expressions, whole clauses, or even larger syntactical units that are then related to specific concept identifiers.[9]

Text mining can be summarised as a process of enumerating text. All words in the input documents are indexed and counted in order to compute a table of documents and words, enumerating the frequency of each word in each document. Subsequently data-mining techniques are applied to derive dimensions or clusters of words or documents, or to identify important words that best predict another variable of interest. Because data mining assumes that data have already been stored in a structured format, much of its pre-processing entails two critical tasks: scrubbing and normalising data and creating extensive numbers of table joins. In contrast, for text-mining systems pre-processing operations focus on identification and extraction of representative features for natural language documents. These pre-processing operations are responsible for transforming unstructured data stored in document collections into a more explicitly structured intermediate format, a concern that is not relevant for most data-mining systems.[9]

Since it is suitable for inferring valuable information from large volumes of unstructured text, text mining has been widely adopted to explore the complex relationships in online discussions. The application of text mining is an effective means for content searches in the textual fields of online discussions.

Text-mining systems architecture

On a functional level, text-mining systems follow the general model provided by some classic data-mining applications, and are roughly divisible into four main areas:

1. pre-processing tasks
2. core mining operations
3. presentation layer components and browsing functionality
4. refinement techniques.

Pre-processing tasks include all those routines, processes and methods required to prepare data for a text-mining system's core knowledge discovery operations. These tasks are typically centred on data source pre-processing and categorisation activities. Pre-processing tasks generally convert the information from each original data source into a canonical format before applying various types of feature extraction methods to create a new collection of documents fully represented by concepts.

Core mining operations are the heart of a text-mining system and include pattern discovery, trend analysis and incremental knowledge discovery algorithms. Amongst the commonly used patterns for knowledge discovery in textual data are distributions (and proportions), frequent and near-frequent concept sets and associations. Core mining operations can also concern themselves with comparisons between and identification of levels of 'interestingness' in some of these patterns.

Presentation layer components include graphical user interface and pattern browsing functionality as well as access to the query language. Visualisation tools and user-facing query editors and optimisers also fall into this architectural category. Presentation layer components may include character-based or graphical tools for creating or modifying concept clusters as well as for creating annotated profiles for specific concepts or patterns.

Refinement techniques, at their simplest, include methods that filter redundant information and cluster closely related data, but may grow in a given text-mining system to represent a full, comprehensive suite of suppression, ordering, pruning, generalisation and clustering approaches aimed at discovery optimisation. These techniques have also been described as post-processing.[10]

The model (Figure 1) is extended by Konchady[11] to text-mining systems that employ pre-processing operations to transform raw, unstructured, original format content into a carefully structured and intermediate data format. Knowledge discovery operations are applied on this specially structured intermediate representation of the original document collection.

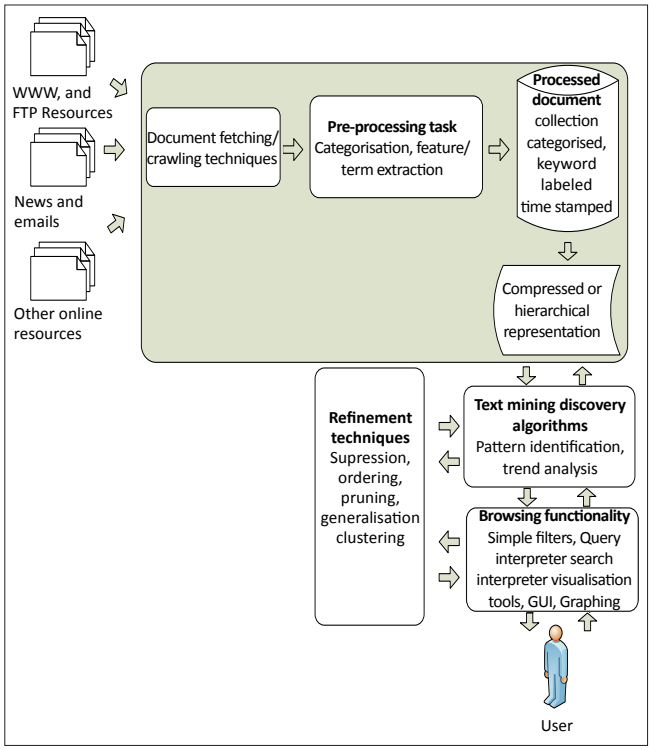
Figure 2 shows a model in which data flow upwards, with an application layer at the top. Input documents are received in the bottom layer, and converted to unstructured text in the standardisation level. The tokenisation layer breaks the stream of text into units called tokens, which can be thought of as a single unit of information. A set of functions in the next layer uses the tokens as input. We do not need the order of tokens to represent a document; the set of unique words occurring in a document or the collection weighted by their importance in the document is sufficient representation. The next layer comprises steps for text mining.[11]

K-means++ clustering algorithm

Text documents as objects to be clustered are very complex and rich in internal structure; hence the documents must be converted into vectors in the feature space to enable clustering. One way of doing this is using 'bag-of-words' document representation, where each word is a dimension in the feature space.

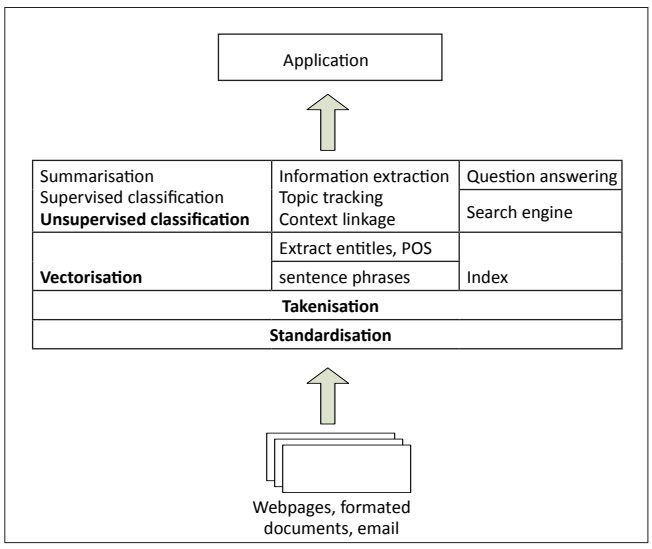
Each vector representing a document in this space will have a component for each word. If a word is not present in the document, the word's component of the document vector will be zero. Otherwise, it will be some positive value, depending on the frequency of the word in the document and in the complete document collection.

We translated our textual data through indexing, normalisation using term frequency-inverse document frequency (TF*IDF) and SVD, into data points for application of k-means++ clustering. The proposed algorithm aims at spreading initial k cluster centres away from each other. The first cluster centre is chosen at random from the data points being clustered, after which each subsequent cluster centre is chosen from the remaining data points, with a probability proportional to its square distance to the point's closest cluster centre. Figure 3 illustrates the algorithm.[12]



Source: Adapted from Feldman R, Sanger J. The text mining handbook; advanced approaches for analyzing unstructured data. New York: Cambridge University Press; 2007

FIGURE 1: System architecture for generic text mining system.



Source: Adapted from Konchady M. Text mining application programming. Boston, Massachusetts: Charles River Media; 2006

FIGURE 2: Layered model of text mining systems.

1a. Choose an initial centre c_j uniformly at random from X.
//X denotes all data points representing term and document vectors.

1b. Choose the next center $C_{i,}$ selecting $c_{i=}\ x \in X$ with probability $\frac{D(x')^2}{\Sigma_{rex} D(x)^2}$

//$D(x)$ denotes the shortest distance from a data point x to the closest center al ready chosen.

1c. Repeat steps 1b, until a total of k centers are chosen.

2. For each $i \in \{1,\cdots, k\}$, set the cluster C_i to be the center of points in X that are closer to c_i than c_j for all $j \neq i$.

3. For each $i \in \{1,\cdots, k\}$, set c_i to be the center of all points in C_i

$$c_i = \frac{1}{|C_i|} \sum xeCi^x$$

4. Repeat Steps 3 and 4 until c_i no longer moves.
// c_i denotes cluster
// c_i denotes cluster centers or centroids.

Source: Adapted from Arthur D, Vassilvitskii S. K-means ++: The advantages of careful seeding. Proceedings of the 18th Annual Association of Computing Machinery-Society for Industrial and Applied Mathematics (ACM-SIAM) Symposium on Discrete Algorithms; 2007 Jan 7–9; New Orleans, Louisiana. Philadelphia: SIAM; 2007. p. 1027–1035

FIGURE 3: K-means++ clustering algorithm.

Proposed method

We propose a new text-mining approach that extracts keywords and clue words from online discussions using k-means++ with latent semantic analysis. We applied SVD to reduce the dimension space and also derived the semantic structure of words appearing in a discussion forum. Online discussions occur in different formats and different languages. Our text corpus consisted of students' postings consisting of non-English words. This introduced a lot of noise into the dataset, resulting from spelling mistakes, abbreviations and acronyms.

After representing each message by its constituent words (terms) and their occurrence as a 'bag of words', we determined which features best described each message, using a TF*IDF normalisation measure. We then determined a posting matrix. In this matrix each index term is a row and each posting is a column. Each cell contains the number of times that terms occur in a posting. Each posting becomes a count vector representing a $|D|$-dimensional vector space. Terms are the axes of the space. We modified the counts with TF*IDF so that rare terms were weighted more heavily than common terms. The resultant matrix $A = [A_1, A_2, \ldots A_n]$, where each column vector A_i represents the weighted term frequency vector of a posting, is highly dimensional and sparsely distributed since most entries are zero. We carried out feature selection using SVD to reduce dimensions by removing irrelevant features and to derive the latent semantic structure from the terms representing the postings.

SVD is used to decompose the terms in document matrices to construct a semantic vector space which can be used to represent conceptual term-document associations. SVD decomposes a matrix into the product of three other matrices. One component matrix describes the original row entities as vectors of derived orthogonal factor values, another describes the original column entities in the same way, and the third is a diagonal matrix containing scaling values, such that when the three components are multiplied, the original matrix can be reconstructed.[9] Because of the orthogonal characteristics of derived factors, terms in a factor have little relation with terms in other factors, but terms in a factor have high correlation with terms in that factor. Thus SVD is able to handle noise resulting from spelling mistakes, abbreviations and acronyms by capturing and modelling inter-relationships amongst terms.

Experiment, results and discussion

Firstly, we extracted the data from the source systems. Our target system was the Moodle course management system, employed for undergraduate students taking a Database Management Systems course. Our data source required minimal manipulation, involving selecting only certain columns to be loaded. The Moodle database had about 198 tables. We only required one table: mdl_forum_posts, that captures the content of messages sent by students. From this table we selected sender, date_sent, and message columns. The structured data included: Sender, Date and Time Sent, amongst others. The unstructured data were the contents of the messages. Message contents were extracted from each posting in the online discussion forum and parsed into sequences of tokens to represent each posting for subsequent analysis. The data dimension was high, with thousands of words representing the online discussion postings. During extraction the data were parsed to remove html tags and other delimiters such as space, tab, new line and other characters. The messages were cleaned by removing delimiters and stop-words and then tokenised and represented as a 'bag of words'. We loaded the data into the end target posting repository.

We created a weighted term-posting matrix that was very large and sparsely populated. We then applied SVD to generate a reduced semantic space.

We considered two singular values of the diagonal matrix to generate term vectors and document vectors after applying SVD (see Table 1 and Table 2). We chose two singular values as these would simplify our representations of these vectors. These values were used as data points from which the k-means++ clustering algorithm was applied.

We run the k-means++ algorithm with latent semantic analysis technique of SVD and one without to generate 2–10 clusters of terms from which we infer the topic of discussion. We reported the clustering validation measure for each.

Table 3 shows how a topic of discussion can be identified from the set of key words in every cluster. We run the algorithm to generate five clusters, and the output was the cue words representing each cluster. Considering cluster four, we can conclude that students were discussing topics related to database systems. This result demonstrates how the proposed method can be used to assess the topics of discussions expressed by students' discussion forums in online learning systems.

TABLE 1: Representation of term vectors.

Nr	Terms	X x 10^{-4}	- Y x 10^{-4}
T1	file	138	35
T2	benefits	13	22
T3	database	81	29
T4	cat	1	8
T5	Dbms	32	71
T6	Unlike	19	11
T7	different	342	7
T8	centralised	19	11
T9	scattered	15	16
T10	dependent	19	11
T11	security	139	34
T12	share	106	84
T13	independence	11	7

TABLE 2: Representation of document vectors.

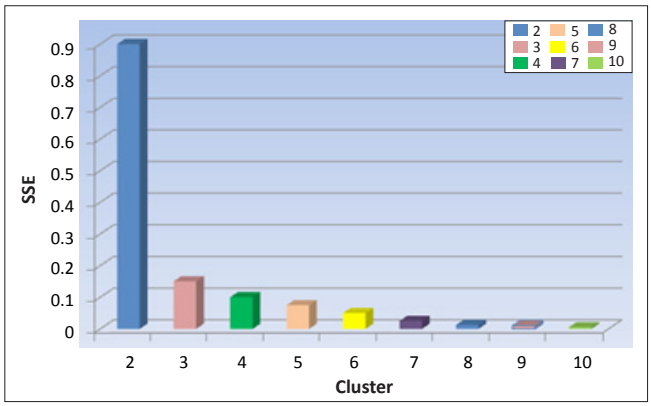
#Post	X x 10^{-4}	- Y x 10^{-4}
P1	2	5
P2	83	44
P3	211	351
P4	9908	269
P5	423	42
P6	56	99
P7	211	149
P8	104	257
P9	34	72
P10	0	3
P11	15	24
P12	16	5
P13	32	19
P14	85	-46
P15	96	-6871

TABLE 3: Top extracted terms in a cluster.

Cluster	Top keywords
0	Photo, Break, Forgot, Bare, Brutal
1	Nice anyone, Real, Mmmmh, Air, John
2	Kaquote, Eggblablabla, Funny, Haha, mellisa
3	Ends, Paperit, ablesformreports, Buying, Write
4	Database, security, Storage, access file

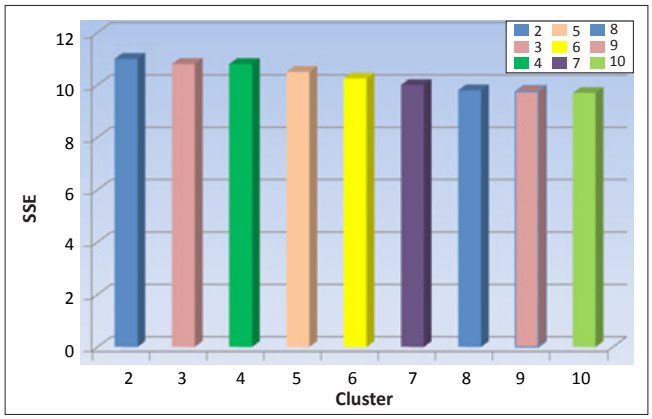
Figure 4 shows that latent semantic analysis helps improve quality of clustering by minimising the sum of squares error. It also helps in minimising inter-cluster similarity and maximising intra-cluster similarity. Figure 5 shows that standard k-means++ clustering has a high sum square-error compared to k-means++ with LSA. Intuitively, as more clusters are added the error function value decreases with both algorithms. From the results we show that a k-means++ clustering algorithm combined with SVD performs better than a stand-alone k-means++, by minimising the sum of squared error objective function. This can be attributed to feature reduction as a result of SVD.

Figure 6 shows that the k-means++ with SVD algorithm generated the final clustering result in a longer period of time; however, as the number of clusters increased, both



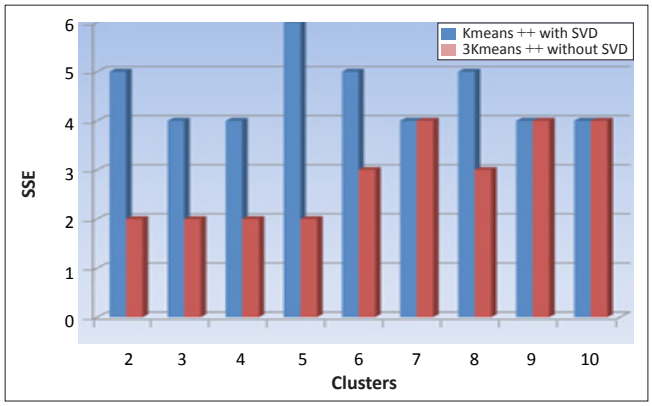
Source: Authors' own data
SSE, sum of squared errors.

FIGURE 4: K-means++ with latent semantic analysis.



Source: Authors' own data
SSE, sum of squared errors.

FIGURE 5: Standard k-means++.



Source: Authors' own data
SVD, singular value decomposition.

FIGURE 6: Comparison of running times.

algorithms had the same running times. This is because k-means++ with SVD uses more time in the dimension reduction process, after which it executes at the same speed as standard k-means++. K-means++ with SVD is expected to execute faster as it works on fewer dimensions after the dimension reduction process.

Conclusion

We propose a hybrid k-means++ algorithm which combines the steps of dimensionality reduction through SVD and a

novel initialisation approach of setting cluster centres. We validated the effectiveness of the proposed algorithm using Moodle course management forum data set by partitioning it into k clusters in such a way that the sum of the clustering errors for all clusters was reduced as much as possible.

For future work we propose use of statistical methods to compute the value of *k*, depending on the data distribution, and an adaptive database to allow the tool to handle free-flow discussions. We also propose parallelisation techniques for SVD to reduce computational loads. SVD can also be combined with feature selection techniques to enhance the accuracy of clustering. Our algorithm will also be tested on larger data sets.

Acknowledgements

We acknowledge the School of Computing and Informatics, University of Nairobi, for allowing us ample access to computing resources whilst carrying out the research.

Competing interests

The authors declare that they have no financial or personal relationship(s) which may have inappropriately influenced them in writing this paper.

Authors' contributions

Y.A. (Kenya Methodist University) designed and conducted experiments in the research and contributed to the analysis and reporting of results. R.O. (University of Nairobi) was the principal researcher and contributed to the design of experiments and analysis and reporting of results.

References

1. Kim J, Shaw E, Feng D, Beal C, Eduard HE. Modeling and Assessing Student Activities in On-Line Discussions. Proceedings of the Association for the Advancement of Artificial Intelligence Workshop; 2006 July 16–17; Boston, MA. Boston: AAAI Press; 2006.
2. Ali S, Salter G. Use of templates to manage online discussion forums. Electronic Journal of e-learning. 2004; 2(1): 11–18.
3. Song W, Park SC. A novel document clustering model based on latent semantic analysis. Proceedings of the Third International Conference on Semantics, Knowledge and Grid; 2007 Oct 29–31, Xi'an, China. Washington: IEEE; 2008. p. 539–542.
4. Wijekumar K, Spielvogel, J. Intelligent Discussion Boards: Promoting Deep Conversation in Asynchronous Discussion Boards through Synchronous Support. Campus-Wide Information Systems. 2006; 22(3): 221–231.
5. Ramampiaro H, Paulsen JR. Combining latent semantic indexing and clustering to retrieve and cluster biomedical information. In: A 2-step approach. Proceedings of the Norsk Informatikk Konferanse (NIK); 2009 Nov 23–25; Trondheim, Norway. Tapir: Akademisk Forlag; 2009.
6. Li N, Wu D. Using text mining and sentiment analysis for online forums hotspot detection and forecast. Elsevier: Decision Support Systems. 2010;48:354–368. http://dx.doi.org/10.1016/j.dss.2009.09.003
7. Stavrianou A, Chauchat, J, Velcin, J. A Content-Oriented Framework for Online Discussion Analysis. Proceedings of 23rd International Conference on Advanced Information Networking and Applications; 2009 May 26–29; Bradford, UK. Available from: http://doi.ieeecomputersociety.org/10.1109/AINA.2009.13
8. Sebastiani F. Machine learning in automated text categorization. ACM Comput. Surv. 2002;34:1–47. http://dx.doi.org/10.1145/505282.505283
9. Landauer, T, Foltz P, Laham, D. Introduction to Latent Semantic Analysis. DPr. 1998;25:259-284. http://dx.doi.org/10.1080/01638539809545028
10. Feldman R, Sanger J. The text mining handbook; advanced approaches for analyzing unstructured data. New York: Cambridge University Press; 2007.
11. Konchady M. Text mining application programming. Boston, Massachusetts: Charles River Media; 2006.
12. Arthur D, Vassilvitskii S. K-means++: The advantages of careful seeding. Proceedings of the 18th Annual Association of Computing Machinery-Society for Industrial and Applied Mathematics (ACM-SIAM) Symposium on Discrete Algorithms; 2007 Jan 07–09; New Orleans, Louisiana. Philadelphia: SIAM; 2007. p. 1027–1035.