MALAY DOCUMENTS CLUSTERING ALGORITHM BASED ON SINGULAR VALUE DECOMPOSITION

Nordianah Ab Samat, Masrah Azrifah Azmi Murad, Muhamad Taufik Abdullah, Rodziah Atan
Faculty of Computer Science and Information Technology, University Putra Malaysia,
43400 Serdang, Selangor, Malaysia
E-mail: nordianahSamat@gmail.com, {masrah, taufik, rodziah}@fsktm.upm.edu.my

ABSTRACT

Document categorization is a widely researched area of information retrieval. A research on Malay natural language processing has been done up to the level of retrieving documents but not to the extent of automatic semantic categorization. Thus, an approach for the clustering of Malay documents based on semantic relations between words is proposed in this paper. The method described in this paper uses Singular Value Decomposition (SVD) technique for the vector representation of each document where familiar clustering techniques can be applied in this space. The experimental results we obtained taking into account the semantics of the document that performed good document clustering by obtaining relevant subjects appearing in a cluster.

Keywords: Singular Value Decomposition (SVD), Latent Semantic Indexing (LSI), Document Clustering, Malay Natural Language Processing

1. INTRODUCTION

An Information Retrieval system typically produced a ranked list of documents in response to a user's query. If the query is general, it is extremely difficult to identify the specific document which the user is interested in. A natural alternative to ranking is to cluster the retrieved set into groups of documents with common subjects. Document clustering is a procedure to separate documents according to certain criteria, for instance documents of different topics.

The volume of digital documents increases rapidly in recent years, therefore an accurate method to categorize large amount of documents are needed imminently. The idea of clustering search results is not new, and has been investigated quite deeply in information retrieval [1,2]. A research on Malay natural language processing has been done up to the level of retrieving documents [3,4] but not to the extent of automation categorization in a semantic nature. In this research, we attempt to use the text mining techniques, i.e. categorization in the context of Malay natural language processing. Nevertheless, it's believed the method build from this research is possible to be used in other languages.

The notion of document similarity between documents is crucial because a document can address multiple area topics. The semantic similarity has many forms and many ways to capture. A popular approach assumes that terms occurring often together in the documents are related to similar topics with high probability. SVD has shown capability of finding such similarities. Thus, this paper proposes a framework to cluster the Malay documents based on SVD.

The paper is organized as follows. In the next section, the related works for document categorization techniques is discussed. Section 3 describes the algorithm to perform document clustering, section 4 describes the datasets of our experiment and section 5 reports on preliminary results and give some examples of the clusters obtained. Finally, section 6 concludes the paper**.**

.

2. RELATED WORK

Document clustering has been traditionally investigated mainly as a means of improving the performance of search engines by pre-clustering the entire corpus [5]. In order to run the clustering algorithm, there have been several different kinds of vector representation in the literature, among them a Vector Space Model (VSM) [6]. In VSM, documents and queries are represented as vectors in term space. Under the vector model, a collection of *n* documents with *m* unique terms is represented as an *m x n* term-document matrix. Although the

VSM is simple and fast, there are a few drawbacks of using the VSM. It cannot reflect similarity of words and only counts the number of overlapping words and it ignores synonymy and polysemy.

Latent Semantic Indexing (LSI) [7] is an information retrieval technique that was designed to address the deficiencies of the classic VSM technique. LSI assumes that there is some underlying or latent structure in word usage that is partially obscured by variability in word choice. LSI extends the vector space model by modeling term-document relationships using a reduced approximation for the column and row space computed by the SVD of the term by document matrix. In fact, the idea of clustering search results using LSI as a means to improve retrieval performance has been investigated quite deeply in Information Retrieval [1,8,9,10,11,12].

The Paraphrase system [8] introduces LSI as a means to cluster search results. The paper focuses on the relationship among clusters produced by LSI and their labels, obtained by taking the most representative words for each cluster. Besides this, the author follows a rather typical approach to select the number *k* of singular values, specifically *k* is fixed and equals to 200. The choice of *k* is critical. Ideally, *k* should be large enough to fit the real structure in the data, but small enough such that noises, sampling errors or unimportant details are not modeled. In fact, values in the range 100-200 were considered as optimal in retrieval applications.

Carrot Search [1,9] is a snippet-based clustering engine. A primary concern is to produce meaningful descriptions for the clusters. To do this, first SVD is performed on a snippet-term matrix to identify a number of relevant topics. Then, phrase analysis is done to identify, for each of the selected topics, a phrase that represents a good description. Finally, documents are assigned to clusters based on the contained phrases. Lingo work is inspired from Grouper [10,11] to analyze short document abstracts, usually contains from 0 to 40 words, and therefore can be analyzed very quickly in clustering step. However, snippets are often hardly representative of the whole document content, and this may in some cases seriously worsen the quality of the clusters.

3. DOCUMENT CLUSTERING

This section outlines techniques involved in our document clustering algorithm.

3.1 Document Preprocessing

The preprocessing basically consists of a process to optimize the list of terms that identify the collection. The aim of the processing phase is to prune from the document all characters and terms with poor information that can possibly affect the quality of group descriptions. Although SVD is capable of dealing with noisy data, without sufficient preprocessing, the majority of discovered abstract concepts would be related to meaningless frequent terms [1].

The first process is by removing stop words. The stop words are frequent words that carry no information and meaningless when used as a search terms (i.e., pronouns, prepositions, conjunctions etc). These words occur too frequently in a document and are usually ignored by the system when searching is done. Stop words may be eliminated using a list of stop words. If a word in the document matches a word in the stop list, then the word will not be included as part of the query processing. An advantage of using stop words is that it could reduce the number of terms that identify the document.

The second process is to stem a word. Morphological variants of words usually have similar meanings. If these words are conflated into a single term, the performance of document retrieval can be improved. This may be done using the process of stemming in such a way that words are stemmed into a root form by removing their affixes. For example, the Malay words *jalan*, *berjalan*, *menjalani*, *dijalankan* dan *perjalanan* are grouped to the root (stem) *jalan*. Precisely, the root of a word is obtained by removing all or some of the affixes attached to the word. The Malay affixes consist of four different types, which are the prefix, suffix, prefix-suffix pair and infix. In this research, the RFO stemmer [13] is used in order to remove the Malay affixes.

3.2 Term-Document Matrix

The next step after preprocessing is to represent documents as vectors in a multidimensional term space [5]. A collection of *d* documents described by *t* terms can be represented as a $t \times d$ matrix A, hereafter referred to as the term-document matrix. It is a sparse matrix whose rows correspond to documents and whose columns correspond to weighted terms in the documents. There are several weighting schemes [5,14] that can be used to construct document vectors. Generally

speaking, the weight w_{ij} of term t_i in document d_j is given by the product of three different factors:

$$w_{ij} = L_{ij} G_i N_j \quad (1)$$

Where L_{ij} is the local weight of term *i* in the document *j*, G_i is the global weight of term *i* in the document collection, and N_j is the normalization factor for document *j*. The following table summarizes the forms of local weight, global weight and normalization weight that have been used in our work. Let us call f_{ij} the frequency of term *i* in document *j*, F_i the global frequency of term *i* in the whole document collection, n_i the number of documents in which term *i* appears, *N* the total number of documents, and *m* the size of a document vector v_j.

Table 1: Term weighting schemes

Weight	*Abbr.*	Formula
Local	SQRT	$\sqrt{f_{ij} - 0.5} + 1$ if $f_{ij} > 0$ 0 if = 0
Global	IGFL	$\log\left(\frac{F_i}{n_i} + 1\right)$
Normalization	COSN	$N_j = \frac{1}{\sqrt{\sum_{i=0}^{m} (G_i L_{ij})^2}}$

In order to get the best and compatible term weighting, we did several experiments for term weighting methods on our data collection. As the results shown in Table 1 we finally used Square Root for local weights, Log-global frequency IDF for global weights and cosine normalization for normalization factor.

3.3 Singular Value Decomposition

This matrix is then analyzed by SVD to derive our particular latent semantic structure model. Singular Value Decomposition (SVD) is a form of factor analysis, or more properly, the mathematical generalization of which factor analysis is a special case [15]. It constructs an *n* dimensional abstract semantic space in which each original term and each original document are presented as vectors.

In SVD a rectangular term-by-document matrix *A* is decomposed into the product of three other matrices *T*, *S*, and *D'* (refer to Figure 1).

$$\{A\} = \{T\}\{S\}\{D\}' \quad (2)$$

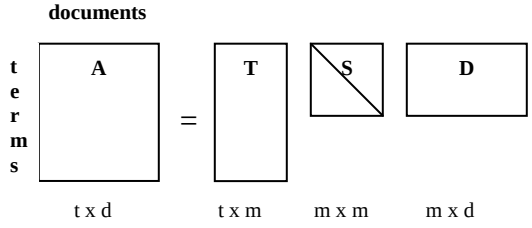

Figure 1: SVD Description

T is an orthonormal matrix and its rows correspond to the rows of *A*, but it has *m* columns corresponding to new, specially derived variables such that there is no correlation between any two columns; i.e., each is linearly independent of the others. *D* is an orthonomal matrix and has columns corresponding to the original columns but *m* rows composed of derived singular vectors. The third matrix *S* is an *m* by *m* diagonal matrix with non-zero entries (called singular values) only along one central diagonal. The role of these singular values is to relate the scale of the factors in the other two matrices to each other such that when the three components are matrix multiplied, the original matrix is constructed.

3.4 Dimension Reduction

Following the decomposition by SVD, the *k* most important dimensions (those with the highest singular values in *S*) are selected. All other factors are omitted, i.e., the other singular values in the diagonal matrix along with the corresponding singular vectors of the other two matrices are deleted. Ideally, *k* should be large enough to fit the real structure in the data, but small enough such that noises, sampling errors or unimportant details are not modeled [7]. The reduced matrix ideally represents the important and reliable patterns underlying the data in *A*. It corresponds to a least-squares best approximation to the original matrix *A*. The optimal of *k* is determined empirically for each collection and is typically around between 200 and 300 for the large documents set. It is therefore apparent that fixing the value of *k*, as it is often done in the literature, would not give good classification performance. The fact that lower

values of *k* are to be preferred in clustering applications is confirmed for example in [16].

In order to calculate the value of *k*, we have chosen the one based on the Frobenius norms of the term- document matrix and its *k*-rank approximation [1]. The selected method requires that a percentage quality threshold *q* be assumed that determines to what extent the *k*-rank approximation should retain the original information. In this way, *k* is set to the minimum value that satisfies the following condition:

$$\frac{\|A_k\|_F}{\|A\|_F} = \frac{\sqrt{\sum_{i=1}^{k}\left(\sigma_i^2\right)}}{\sqrt{\sum_{i=1}^{r_A}\left(\sigma^2 i\right)}} \geq q \qquad (3)$$

In the above formula, A is the original term-document matrix, A_k is its *k*-rank approximation, r_A is the rank of the A matrix, σ_i is its *i*th singular value (the *i*th diagonal element of the SVD's $\sum$ matrix) and $\|A\|_F$ denotes the Frobenius norm of the A matrix. Clearly, the larger the value of q the more cluster label candidates will be induced. The choice of the optimal value for this parameter ultimately depends on the users' preferences.

3.5 Clustering

The next step is to run a clustering algorithm in this reduced space to cluster documents with respect to their topics. In our research, we use k-means algorithm to cluster our document collection into a few tightly structured ones.

K-means is an iterative algorithm in which clusters are built around *n* central points called centroids [1]. The number of clusters, *n* is determined using the same value of *k* previously in dimension reduction. A natural intuition suggests that, assuming the document collection contains *n* hidden clusters, the natural value of *k* to be used for SVD is exactly *n* [12]. It assumes each cluster in the vector space informally corresponds to some clearly defined topic. The algorithm starts with a random set of centroids and assigns each document vector to its closest centroid. Then, repeatedly, for each group, based on its members, a new central point (new centroid) is calculated and object assignments to their closest centroids are changed if necessary. The algorithm finishes when no object reassignments are needed or when certain amount of time elapses (refer to Figure 2).

Inputs :
$A = \{a_1 \ldots a_k\}$ (Document vectors to be clustered)
n (Number of clusters)

Outputs :
$C = \{c_1 \ldots c_n\}$ (cluster centroids)
$m : A \rightarrow \{1 \ldots n\}$ (cluster membership)

Procedure K-Means

Set C *to initial value (random selection of* A*)*
For each $a_i \in A$
m (a_i) = *arg min distance* (a_i,c_j)
$j \in \{1...n\}$
End
while m *has changed*
For each $i \in \{1...n\}$
Recompute ci as the centroid of {a|m(a) = i}
End
For each $a_i \in A$
m (a_i) = *arg min distance* (a_i,c_j)
$j \in \{1...n\}$
End
End
End

Figure 2: K-Means Algorithm

4. DESCRIPTION OF DATASETS

In order to construct datasets, we ran Malay queries on Google and selected a number of the top-ranked search results. Those results were manually classified into a number of clusters. Then, the algorithm was run on the document collection to compare the suggested clusters with those identified manually. The manual classification step is necessary to assess the quality of the clustering. As a consequence, document collections tend to be quite small. For query matching, we used a short query, 'perasaan' (feeling) and the relevant results are composed of about 90 documents.

5. RESULTS AND DISCUSSION

We have performed preliminary experiments to illustrate some of the potential benefits of the above approach. The benchmark used to compare the results obtained is based on the traditional VSM. Table 2 and Table 3 show the resulting document clusters using a query 'perasaan' (feeling).

Table 2: Clusters results using SVD

Cluster results for the query : 'perasaan' (feeling) Documents: 90, Clusters: 9		
Cluster Num.	*Size*	**Topics** and sample document titles
1	9	**cinta (love)** 1. Ragam orang bercinta (Behavior after love) 2. Cinta menurut pandangan Islam (An Islamic perspective on love)
2	9	**gembira (happy)** 1. Pilih untuk gembira (Choose to be happy) 2. Detik gembira (Happy moments)
3	9	**benci (hate); marah (angry)** 1. Kenapa anda merasa marah? (Why do you get angry?) 2. Simptom dan gejala penyakit marah (Symtomps of anger)
4	11	**kecewa (dissapointed)** 1. Bagaimana mengatasi kekecewaan? (How to overcome disappointment?) 2. Kekecewaan.. jangan biarkan berlarutan (Dissapoinment.. don't
5	10	**rindu (missing somebody)** 1. Rindu emak (Miss my mom) 2. Kenapa mesti ada rindu (Why do I miss you)
6	11	**cemburu (jealous)** 1. Cemburu betulkah tanda sayang (Is it true jealousy means you are in love?) 2. Bintang jua cemburu (Celebrity also has some feeling of jealousy)
7	11	**sedih (sad)** 1. Apabila titisnya air mata seorang lelaki (when a man crying) 2. Kenapa sedih? (Why so sad?)
8	12	**takut (fear)** 1. Perasaan takut di kalangan kanak-kanak (Fear among kids) 2. Perasaan takut pada tuhan (Fear of god)
9	8	**marah (angry)** 1. Ghadiba (Irate) 2. Ubat marah (The cure for anger)

Table 3: Clusters results using VSM

Cluster results for the query : 'perasaan' (feeling) Documents: 90, Clusters: 9		
Cluster Num.	*Size*	**Topics** and sample document titles
1	11	**sedih (sad); kecewa (disappointed); marah (angry)** 1. Apabila titisnya air mata seorang lelaki (When a man crying) 2. Cepat berang (Get angry easily)
2	8	**sedih (sad); kecewa (disappointed); takut (fear); marah (angry)** 1. Takutilah dosa-dosa batin (Be afraid to sin) 2. Ghadiba (Irate)

3	10	**marah (angry)** 1. Marah cetuskan konflik dan peperangan (Anger is the reason for conflict and war) 2. Ubat marah (The cure for anger)
4	6	**gembira (happy)** 1. Pilih untuk gembira (Choose to be happy) 2. Detik gembira (Happy moments)
5	11	**kecewa (disappointed); marah (angry)** 1. Bagaimana mengatasi kekecewaan? (How to overcome disappointment?) 2. Kenapa anda merasa marah? (Why do you get angry?)
6	9	**cinta (love)** 1. Ragam orang bercinta (Behavior after love) 2. Cinta menurut pandangan Islam (An Islamic perspective on love)
7	14	**takut (fear)** 1. Perasaan takut di kalangan kanak-kanak (Fear among kids) 2. Peasaan takut pada tuhan (Fear to god)
8	11	**rindu (missing somebody)** 1. Rindu emak (Miss my mom) 2. Kenapa mesti ada rindu (Why do I miss you)
9	10	**cemburu (jealous)** 1. Cemburu betulkah tanda sayang (is it true jealousy means you are in love?) 2. Bintang jua cemburu (Celebrity also has some feeling of jealousy)

From the results shown in Table 2 and Table 3 above, SVD managed to produce groups or clusters that contain similar topics in them. Documents belonging to the same clusters are similar to each other, while documents from two different clusters are dissimilar. For example, documents with subject on 'feeling happy' are grouped together while documents with subject on 'feeling angry' are placed in a different group. Unlike SVD, three clusters produced by VSM contain documents with multiple topics. The cluster separation ability and topic boundary of VSM can be seen ambiguously. As the results shown, SVD method offers the best performance advantages in document clustering compared to a traditional VSM.

The experimental results we obtained taking into account the semantics of the document. SVD retrieves only semantically similar documents with documents on similar topics being clustered together. It is because SVD examines the document collection as a whole, to see which other documents contain some of those same words. SVD considers documents that have many words in common to be semantically close and ones with few words in common to be semantically distant. As the example, we could infer documents that contain a word 'air mata' (tears) is a kind of feeling sad even though they do not share a word 'sedih' (sad). This is because in other documents, a word 'sedih' (sad) tends to occur in the same context as a word 'air mata' (tears). Unlike SVD, most document clusters produced by VSM are plausible although some documents may be dissimilar.

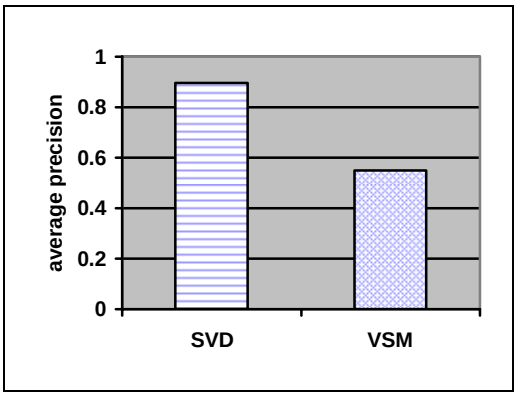

Figure 3: Average precision of SVD compared to VSM

To visualize the difference, Figure 3 shows VSM yields lower average precision. If we take average precision to compare overall performance of the similarity measures, SVD is superior to

VSM +63.55%. As the results show, SVD method offers improvement over the popular VSM method.

6. CONCLUSION

We have presented a framework to document clustering based on SVD in the context of Malay natural language processing. We perform good document clustering by obtaining similar subjects appearing in a cluster. Preliminary experimental evaluations show that the SVD approach leads to dramatic dimension reduction while achieving good clustering results compared to VSM.

We are currently continuing comparison with other techniques and testing our system with a large and different document collection to ensure that it will produce a consistently satisfactory result.

In future, we plan to extract the label description for each cluster using a few phrases that provide the user an overview of topics covered in the document clusters. This is to help the user better understand the information contained in each document cluster, hence, the user may save time and identify the specific group of documents they are looking for.

REFRENCES

[1] S. Osinski, J. Stefanowski, D.Weiss, "Lingo: Search results clustering algorithm based on singular value decomposition," *Proceedings of the International Conference on Intelligent Information Systems (IIPWM)*, 2004.

[2] Shankaran Sitarama, Uma Mahadevan and Mani Abrol, "Efficient cluster representation in similar document search," *Proceedings of the Twelfth International World Wide Web Conference*, WWW2003, Budapest, Hungary, 2003.

[3] M.P. Hamzah and T.M.T. Sembok, "Enhancing retrieval effectiveness of Malay documents by exploiting implicit semantic relationships between words," *Transactions on Engineering, Computing and Technoloy*, V10, ISSN 1305-5313, 2005.

[4] S.M.F.D S Mustapha, N. Idris and R. Abdullah, "Embedding information retrieval and nearest-neighbor algorithm into automated Essay Grading System," *Third International Conference on Information Technolgy and Applications*, volume 2, 2005, pp. 169-172

[5] Van Rijsbergen, C.J., Information Retrieval, 2nd edition, Butterwoth 1979.

[6] Baeza-Yates, R., & Ribeiro-Neto, B. Modern Information Retrieval, ACM Press, 1999.

[7] Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., & Harshman, R., "Indexing by Latent Semantic Analysis," *Journal of the American Society For Information Science*, 41, 1990, 391-407/

[8] P.Anick, S. Vaithyanathan, "Exploiting clustering and phrases for context-based information retrieval," *ACM SIGIR*, 1997, pp. 314-323.

[9] S. Osinski, D. Weiss, "A concept-driven algorithm for clustering search results," *IEEE Intelligent Systems* 20 (3), 2005, 48-54.

[10] O. Zamir, O.Etzioni, "Web document clustering: A feasibility demonstration," *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1998.

[11] O. Zamir, O. Etzioni, "Grouper: a dynamic clustering interface for web search results," *Computer Networks* 31 (11-16), 1999, 1361-1374.

[12] G. Mecca, S. Raunich, A. Pappalardo, "A new algorithm for clustering search results," *Data & Knowledge Engineering* 62, 2007, 504-522.

[13] Abdullah M.T., Ahmad F., Mahmud R., and Sembok T.M.T., "A stemming algorithm for Malay language," *Proceedings of the 4th International Conference on Information Technology in Asia*, Kuching, Malaysia, 2005.

[14] E. Chisholm, T.G. Kolda, "New term weighting formulas for the vector space method in information retrieval," *Technical Report No. ORNL/TM-13756, Computer Science and Mathematics Division* – Oak Ridge National Laboratory, 1999.

[15] Berry, M. W., Dumais, S.T., & O'Brien, G.W., "Using linear algebra for intelligent information retrieval," *SIAM Reviews*, 37, 1995, 73-595.

[16] H. Schutze, C. Silverstein, "Projections for efficient document clustering," *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1997.