Categorization of Malay Documents using Latent Semantic Indexing

Nordianah Ab Samat, Masrah Azrifah Azmi Murad,
Rodziah Atan, Muhammad Taufik Abdullah

Faculty of Computer Science and Information Technology
Universiti Putra Malaysia, 43400 Serdang, Selangor
E-mail : nordianahSamat@gmail.com

Faculty of Computer Science and Information Technology
Universiti Putra Malaysia, 43400 Serdang, Selangor
Tel : 03-89466546, Fax : 03-89466576
E-mail : masrah@fsktm.upm.edu.my

Faculty of Computer Science and Information Technology
Universiti Putra Malaysia, 43400 Serdang, Selangor
Tel : 03-89466574, Fax : 03-89466576
E-mail : rodziah@fsktm.upm.edu.my

Faculty of Computer Science and Information Technology
Universiti Putra Malaysia, 43400 Serdang, Selangor
Tel : 03-89466529, Fax : 03-89466576
E-mail : taufik@fsktm.upm.edu.my

ABSTRACT

Document categorization is a widely researched area of information retrieval. A popular approach to categorize documents is the Vector Space Model (VSM), which represents texts with feature vectors. The categorizing based on the VSM suffers from noise caused by synonymy and polysemy. Thus, an approach for the clustering of Malay documents based on semantic relations between words is proposed in this paper. The method is based on the model first formulated in the context of information retrieval, called Latent Semantic Indexing (LSI). This model leads to a vector representation of each document using Singular Value Decomposition (SVD), where familiar clustering techniques can be applied in this space. LSI produced good document clustering by obtaining relevant subjects appearing in a cluster.

Keywords

Latent Semantic Indexing, Document Clustering, K-means, Malay Language

1.0 INTRODUCTION

Information retrieval can be defined broadly as the study of how to determine and retrieve from a corpus of stored information the portions which are relevant to particular information needs (Van Rijsbergen, 1979). The goal of an information retrieval system is to locate relevant documents in response to a user's query at the same time retrieving as few as possible of the irrelevant documents. In order to represent the documents efficiently, those with similar topics or contents are clustered together. Categorization will group similar documents together based on their dominant features. The idea of clustering search results is not new, and has been investigated quite deeply in information retrieval (Osinski, Stefanowski, Weiss, 2004; Shankaran, Uma, Mani, 2003) based on the so called cluster hypothesis according to which clustering may be beneficial to users of an information retrieval system since it is likely that results that are relevant to the user are close to each other in the document space, and therefore tend to fall into relatively few clusters.

A research on Malay natural language processing has been done up to the level of retrieving documents (Hamzah & Sembok, 2005a) but not to the extent of automation categorization in a semantic nature. Thus, this paper proposes a framework to document clustering using latent semantic indexing (Deerwester, Dumais, Fumas, Landauer & Harshman, 1990) 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 paper is organized as follows. In the next section, the related works for document categorization techniques is discussed. Section 3 describes the LSI method that was designed to overcome the deficiencies of the classic vector space model, section 4 describes the algorithm to perform document clustering and section 5 reports on preliminary results and give some examples of the clusters obtained. Finally, section 6 concludes the paper.

2.0 RELATED WORK

There have been several different kinds of document categorization in the literature, among them a vector space model (Baeza & Ribeiro, 1999).

2.1 Vector Space Model

The VSM allows searching using natural language. This method proved to be better than Boolean model in grouping similar documents together by using keyword matching. Documents and queries are represented as vectors in term space. The formula for calculating weight is given by:

$$w_{ik} = tf_{ik} * \log (N/df_k) \qquad (1)$$

where tf_{ik} is the frequency of k_{th} term in document i, df_k is the number of documents in which a word occurs and N is the total number of documents in the collection. A query, q is usually expressed in natural language, some restricted form of natural language or a set of keywords. The cosine measure(Baeza & Ribeiro, 1999) is used to measure the angle between two vectors, i.e., a document d_j and a user's query, q. The degree of similarity of the document d_j with regard to the query q can be measured by the cosine of the angle between these two vectors, i.e.:

$$sim(d_j,q) = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w^2{}_{i,j}} \times \sqrt{\sum_{j=1}^{t} w^2{}_{i,q}}} \qquad (2)$$

Many document retrieval systems use the VSM technique because it is simple and fast. However, there are a few drawbacks of using the VSM. For example, VSM cannot reflect similarity of words and only counts the number of overlapping words and it ignores synonymy and polysemy. In synonymy, different writers use different words to describe the same idea. Thus, a person issuing a query in a search engine may not retrieve a relevant document just because that document does not contain the same words which appeared in the query. For example, *taxi* is similar to *cab* and *projector* is similar to *beamer*. In polysemy, the same word can have multiple meanings, so a searcher may get irrelevant documents containing the words he searched for with an alternate meaning. For example, a botanist and a computer scientist looking for the word *tree* possibly look for a different set of document, and a word *domino* could be a *pizza company* or a *game*.

3.0 LATENT SEMANTIC INDEXING

Latent Semantic Indexing (LSI) is an information retrieval technique that was designed to address the deficiencies of the classic VSM technique. LSI tries to remove the problems of lexical matching by using statistically derived conceptual indices instead of individual words for retrieval. LSI assumes that there is some underlying or latent structure in word usage that is partially obscured by variability in word choice. A truncated Singular Value Decomposition (SVD) is used to estimate the structure in word usage across documents. Retrieval is then performed using a database of singular values and vectors obtained from the truncated SVD.

LSI is a variant of the vector space model that converts a representative sample of documents to a term-by-document matrix in which each cell indicates the frequency of each term (rows) occurs in each document (columns). Thus a document becomes a column vector and can be compared with a user's query represented as a vector of the same dimension. 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

3.1 Singular Value Decomposition

Singular Value Decomposition (SVD) is a form of factor analysis, or more properly, the mathematical generalization of which factor analysis is a special case (Berry et al., 1995). It constructs an n dimensional abstract semantic space in which each original term and each original (and any new) 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).

$$\{X\} = \{T\}\{S\}\{D\}' \qquad (3)$$

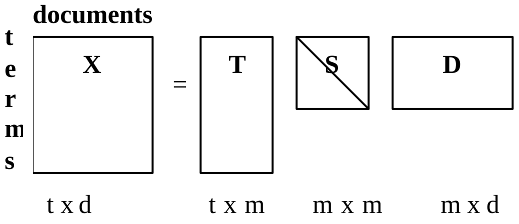

Figure 1 : SVD description

T is a orthonormal matrix and its rows correspond to the rows of X, 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 singualr values) only along one central diagonal. A large singular value indicates a large effect of this dimension on the sum squared error of the approximation. 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.

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. The amount of dimensionality reduction, i.e. the choice of *k* is critical and an open issue in the factor analytic literature. Ideally, *k* should be large enough to fit the real structure in the data, but small enough such that noise, sampling errors or unimportant details are not modeled (Deerwester et al., 1990).

The reduced dimensionality solution then generates a vector of *k* real values to represent each document. The reduced matrix ideally represents the important and reliable patterns underlying the data in *X*. It corresponds to a least-squares best approximation to the original matrix *X*.

4.0 DOCUMENT CLUSTERING

This section outlines steps involved in a document clustering algorithm.

4.1 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 (Osinski, Stefanowski & Weiss, 2004).

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..

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 (Abdullah, Taufik, Ahmad, Mahmud & Sembok, 2005) is used in order to remove the Malay affixes.

The new Malay stemming algorithm can be described as follows:

```
Step-1:  Get the next word until the last word;
Step-2:  Check the word against the dictionary; if it
         appears in the dictionary, the word is the root
         word and goto Step-1;
Step-3:  Get the next rule; if no more rules available,
         the word is considered as a root word and goto
         Step-1;
Step-4:  Apply the rule on the word to get a stem;
Step-5:  Perform recoding for prefix spelling
         exceptions and check the dictionary;
Step-6:  If the stem appears in the dictionary, the stem
         is the root of the word and goto Step-1;
         else goto Step-7;
Step-7:  Check the stem from Step-4 for spelling
         variations and check the dictionary;
Step-8:  If the stem appears in the dictionary, the stem
         is the root of the word and goto Step-1;
         else goto Step-9;
Step-9:  Perform recoding for suffix spelling
         exceptions and check the dictionary;
Step-10: If the stem appears in the dictionary, the stem
         is the root of the word and goto Step-1;
         Else goto Step-3
```

4.2 Term-Document Matrix and SVD Decomposition

The next step after preprocessing is to represent the text as a matrix which describes the occurrences of terms in documents. A collection of *d* documents described by *t* terms can be represented as a $t \times d$ matrix *X*, hereafter referred to as the term-document matrix. It is a sparse matrix whose rows correspond to documents and whose columns correspond to stemmed terms that appear in the documents. Each element *aij* of the term-document matrix represents the frequency of term *i* that appear in the document *j*. This matrix is then analyzed by SVD to derive our particular latent semantic structure model. In SVD, a rectangular matrix *X* is decomposed into the product of three other matrices *T*, *S* and *D*. *T* and *D* is an orthonomal matrix while S is a diagonal matrix.

4.3 Dimension reduction

After the decomposition by SVD, the dimension of matrices should be reduced. The optimal of *k* is determined empirically for each collection and is typically around between 200 and 300 for the large documents set. For our research, the choice of *k* is smaller because currently we used only small set of documents.

4.4 Query matching

The next step is to match a user query with a set of documents in order to gather a number of relevant documents. A user query is represented by a vector in the column space of the term-document matrix. This means that the query can be treated as a pseudo-document that is built solely of the query terms. In order to compare a query or pseudo-document, q, to other documents, we need to be able to start with its term vector X_q and derive a representation D_q. A little algebra shows that:

$$D_q = X_q^{'} T S^{-1} \qquad (4)$$

Now, the new coordinates of the vectors for pseudo-document are obtained.

Therefore, in the process of query matching, documents must be selected whose vectors are geometrically closest to the query vector. A common measure of similarity between two vectors is the cosine of the angle between them. In a $t \times d$ term-document matrix X, the cosine between document vector wij and the query vector q can be computed according to the formula (refer to equation 2).

The similarity value therefore has a domain of [-1,1], with the value 1 being "exactly" similar. Documents are sorted in descending order according to the similarity score. Only a group of documents whose cosine similarity is higher then document similarity threshold will be retained for document clustering in the next step.

4.5 Clustering

From the documents whose similarity are higher then Document Similarity Threshold, we run a clustering algorithm in this 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 (Osinski, Stefanowski, Weiss, 2004). 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 $(a_i, c_j)$
      $j \in \{1 \ldots n\}$
  End
  while $m$ has changed
      For each $i \in \{1 \ldots n\}$
        Recompute ci as the centroid of {a|m(a) = i}
      End
      For each $a_i \in A$
        $m(a_i)$ = arg min distance $(a_i, c_j)$
                $j \in \{1 \ldots n\}$
      End
  End
End
```

Figure 2 : K-Means Algorithm

5.0 RESULTS AND DISCUSSION

We have performed preliminary experiments to illustrate some of the potential benefits of the above approach. The collection used for the experiments contains news selected randomly from a Bernama website, composed of about 9 documents (refer to figure 3) and comprising approximately 37 words after stemmed.

D1 : Shahrulhaizy Mahu Catat Hattrick Jalan Kaki Sukan SEA.
D2 : Malaysia kekal kuasai acara jalan kaki Sukan SEA.
D3 : Hoki hampir pasti rangkul emas di Sukan Korat, kata jurulatih.
D4 : Acara jalan kaki Sukan Korat: Malaysia Hanya Mampu Perak.
D5 : Jadual Pertandingan Atlet Malaysia pada Sukan Sea Korat.
D6 : Sukan SEA : Hoki Malaysia raih pingat emas
D7 : Malaysia Pada Kedudukan Untuk Dua Emas Hoki di Sukan SEA.
D8 : Angkasakan Tamadun Bangsa Melayu, kata Najib.
D9 : Malaysia Ditawar Beli Soyuz Bawa Sheikh Muszapahar.

Figure 3 : Document titles

This led to a (37 x 9) term-document matrix of co-occurrences, stored in sparse fashion. We performed the SVD of this matrix and the number of singular values retained was set to 9, which seemed to achieve an adequate balance between reconstruction error and noise suppression. For query matching, we used a short query, i.e., 'Sukan' and find the new coordinates for this pseudo-document. Documents are sorted in descending order according to the similarity score. Document similarity threshold was set to 0.1 and only a group of documents whose cosine similarity is higher then 0.1 were retained (refer to table 1).

Table 1: Document similarity in descending order

Document	Similarity
D3	0.590477
D1	0.541433
D5	0.398382
D4	0.244653
D6	0.207951
D7	0.207951
D2	0.122367

As can be seen from table 1 above, only a group of documents similar to pseudo-document "Sukan" were retained. Two documents whose topics are not relevant were removed. We clustered the vectors in this space into 3 clusters of approximately 3 vectors each using simple K-means algorithm (refer to table 2).

Table 2 : Document clustering using a query 'Sukan'. Clusters are indicated by the dotted lines

Document	General Subject
D1	*Sukan Jalan Kaki*
D2	*Sukan Jalan Kaki*
D4	*Sukan Jalan Kaki*
D3	*Sukan Hoki*
D6	*Sukan Hoki*
D7	*Sukan Hoki*
D5	*Jadual Sukan SEA*

Table 2 shows only relevant subjects appearing in each cluster. Documents belonging to the same cluster are "similar" to each other, while documents from two different clusters are "dissimilar".

6.0 CONCLUSIONS

We have presented a framework to document clustering based on LSI in the context of Malay natural language processing. We perform good document clustering by obtaining relevant subjects appearing in a cluster. In a meantime we did not performed a comparison with other techniques yet but we are certain that this method will produce results better than traditional VSM.

We are currently continuing comparison with previous 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.

REFERENCES

Abdullah, M.T., Ahmad, F., Mahmud, R., and Sembok, T.M.T. (2005). A Stemming Algorithm for Malay Language. *In Proceedings of the 4 th International Conference on Information Technology in Asia 2005. Kuching, Malaysia.*

Baeza-Yates, R. And Ribeiro-Neto. (1999). *B. Modern Information Retrieval, ACM Press.*

Berry, M. W., Dumais, S.T., & O'Brien, G.W. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review, 37,* 73-595.

Deerwester, S., Dumais, S.T., Fumas, G.W., Landauer, T.K., & Harshman, R. (1990). Indexing By Latent Semantic Analysis. *Journal of the American Society For Information Science, 41,* 391-407.

M.P. Hamzah and T.M.T. Sembok. (2005). Enhancing Retrieval Effectiveness of Malay Documents by Exp loiting Implicit Semantic Relationships Between Words. *Transactions on Engineering, Computing and Technology, V10,* ISSN 1305-5313.

M.P. Hamzah and T.M.T. Sembok. (2005). On Retrieval Performance of Malay Textual Documents. *Proceedings of the IASTED Conference on Artificial Intelligence and Applications, Innsbruck, Austria, Feb 13-16,* Track 502-124.

Muhamad Taufik Abdullah, Fatimah Ahmad, Ramlan Mahmod, and Tengku Mohd. Tengku Sembok. (2003). Application of Latent Semantic Indexing on Malay-English Cross Language Information Retrieval. *Proceedings of the 6 th. International Conference on Asian Digital Libraries (ICADL 2003), Kuala Lumpur, Malaysia, 8-12 December 2003, Lecture Notes in Computer Science (LNCS 2911),* pp: 663 – 665.

Shankaran Sitarama, Uma Mahadevan and Mani Abrol. (2003). Efficient Cluster Representation in Similar Document Search. *In Proceedings of the Twelfth International World Wide Web Conference, WWW2003, Budapest, Hungary.*

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

Van Rijsbergen, C.J. (1979). *Information Retrieval, 2nd edition, Butterworth.*