IMPROVING THE RELEVANCY OF DOCUMENT SEARCH USING THE MULTI-TERM ADJACENCY KEYWORD-ORDER MODEL

***Lim Bee Huang*[1], *Vimala Balakrishnan*[2], *Ram Gopal Raj*[3]**
[1,2] Department of Information System,
[3] Department of Artificial Intelligence,
Faculty of Computer Science and Information Technology,
University of Malaya, 50603, Kuala Lumpur
Email: [1]bhlimum@um.edu.my, [2]vimala_balakrishnan@um.edu.my, [3]gopalraj@tm.net.my

ABSTRACT

This paper presents an enhanced vector space model, Multi-Term Adjacency Keyword-Order Model, to improve the relevancy of search results, specifically document search. Our model is based on the concept of keyword grouping. The keyword-order relationship in the adjacency terms is taken into consideration in measuring a term's weight. Assigning more weights to adjacency terms in a query order results in the document vector being moved closer to the query vector, and hence increases the relevancy between the two vectors and thus eventually results in documents with better relevancy being retrieved. The performance of our model is measured based on precision metrics against the performance of a classic vector space model and the performance of a Multi-Term Vector Space Model. Results show that our model performs better in retrieving more relevant results based on a particular search query compared to both the other models.

Keywords: *Information retrieval, keyword-order, vector space model, adjacency terms relationship.*

1.0 INTRODUCTION

The amount of available textual information in electronic form has grown rapidly over time. Thus, the need to find relevant information from large collection of texts has become a challenging issue to many users and technology developers. The traditional information retrieval system is based on a ''one-fits-all'' principle of exact matching behaviour, which results in huge amount of documents to be returned to the users. The users are then required to browse through the returned results in order to find the documents that are relevant to their information needs. Past research has shown that users generally browse through the first 10 or 20 results [1], therefore, ranked results are necessary so that the most relevant documents are listed at the top of pages.

The vector space model is one of the earliest solutions proposed by [2]. Many enhancements to the vector space model have been developed in the past including the study based on relevance feedback [3], terms correlation using data mining technique [4], and adjacency terms relationship [5], among others. These enhancement models have significantly improved the relevance ranking, particularly in terms of providing more relevant results in information retrieval. We propose that the model presented by [5] which is based on adjacency terms relationship, can be further enhanced so that a higher degree of relevant results are produced. The current study aims to do so by emphasizing the adjacency terms and keyword-order relationship. Our proposed model, the Multi-Term Adjacency Keyword-Order (MTAKO) model is based on keyword grouping concept, [6], however, the keyword-order relationship in the adjacency terms will be taken into consideration as well. MTAKO focuses on improving the precision (the proportion of the retrieved documents that are relevant) of the system when responding to multi-term queries.

2.0 VECTOR SPACE MODELS

Vector space model is a statistical model that models documents and queries as vectors in a multi-dimensional space, [2]. The relevancy of a document is judged statistically by computing the cosine of the angle between the document vector and query vector. When the distance between document and query vectors is short in the document space, they are conceptually similar and relevant to the users, [7]. However, the representations of terms in classic vector space model are assumed to be mutually independent, resulting in a loss of term location (position) in the document, [5]. Therefore, a document with two query terms "information" and "retrieval" that are far-apart in location is considered as relevant as a document with query terms that occur next to each other, such as "information retrieval". [5] showed that query terms that appear close to each other constitute more relevant documents than far-apart terms in the document.

3.0 THE ADJACENCY KEYWORDS GROUPING CONCEPT

According to [6], a document may contain keywords found either as standalone (not next to any other keywords) or in groups with other adjacency keywords. For example, a search query of "information retrieval system", and the adjacency terms in a document that possibly found to be keywords are as shown in Table 1.

Table 1: Possible keywords in document for "information retrieval system"

Group Type	Adjacent Keywords
Standalone keywords	"information", "retrieval", "system"
Group with 2 keywords	"information retrieval", "information system", "retrieval system", "system information", "system retrieval", "retrieval information"
Group with 3 keywords	"information retrieval system", "retrieval system information", "system retrieval information", "retrieval information system"

With the search query of three keywords, the maximum size of the keyword group is equivalent to the size of the query keywords, which, is three in the above example. In general, for a search query of *N* keywords, it will contain a maximum of *N* keyword groups. If the number of adjacent keywords, *n* is more than *N*, repeated keywords will occur in the group (e.g. information retrieval system retrieval). In this case, the repeated keyword (i.e. retrieval) is treated as a non-keyword in the query processing. The idea of grouping the keywords in a document is motivated by the assumption that terms found in keyword group should be more significant to the document and given more weight than standalone terms, [6].

4.0 MULTI-TERMS ADJACENCY KEYWORDS ORDER (MTAKO) MODEL

The proposed model in this study is based on two main assumptions:

Assumption 1: If a document contains a keyword group and the terms within the group are in the order defined by the query, this document is assumed to be more relevant than a group without any keyword-order within them.

Assumption 2: The higher the number of keywords in the query order (measured in keyword-order pairs) within the group, the higher the relevancy of the document.

For example, assume a search query contains the keywords in the order of {information, retrieval, system, performance, evaluation} and with four documents in the collection as below:

Doc 1: contains {information, retrieval, system, performance}

Doc 2: contains {information, retrieval, performance, system}

Doc 3: contains {information, system, performance, evaluation}

Doc 4: contains {retrieval, performance, information, evaluation}

The keyword-order pairs found within the group in the documents are shown in Table 2.

Table 2: Keyword order pairs

Document	Keyword Order Pairs
Doc 1	information-retrieval, retrieval-system, system-performance
Doc 2	information-retrieval
Doc 3	system-performance, performance-evaluation
Doc 4	<None>

Based on the assumptions, *Doc 1* has three keyword-order pairs in the keyword group and it is assumed to be the most relevant, followed by *Doc 3* and *Doc 2*. The least relevant document would be *Doc 4* that has no keyword-order pairs within the keyword group.

If any two consecutive terms in a pair within the group match the keyword in the order defined by the query, this group is then considered a keyword-order group. It is then assigned more weight to show the importance of terms that appear in the keyword-order group. The number of keyword-order pairs is incorporated into the weight function to compute the term weight according to the size of the keyword group, the group order and the number of keyword-order pairs in which the term occurs.

5.0 MTAKO VECTOR PROCESSING

Consider a sample text documents collection with query illustrated as in Table 3.

Table 3: Sample document collection

Query: Shipment of gold and silver

Doc No	Content
D1	Shipment of gold and silver damaged in a fire
D2	Order of gold and silver delayed in a shipment
D3	Shipment of silver and gold arrived in a truck

The search process begins with scanning the first document *D1* sequentially from the first term to the last. Then for each term, the term that is found to be a keyword will be added to the keyword array (the purpose of the keyword array is to group all the adjacency keywords together). To construct the document vector, the information needed are terms, keyword group size, keyword group order, term frequency and number of keyword-order pairs that exist. Term frequency is accumulated each time the term occurs in the document. Keyword group size is computed based on the size of the keyword array, whereas the keyword group order and number of keyword-order pairs are determined based on the pseudo-code shown in Fig 1.

```
Input: keyword array, query array
Output: keyword group order, keyword-order pairs
Begin
...
  For each term in keyword array
    Get the position in query array where term matches keyword
    Get next term in keyword array
    Get next keyword in query array
    Compare next term with next keyword
     If next term = next keyword,
       Set keyword group order = 1
       Increment keyword-order pair by 1
...
 End
```

Fig. 1. Pseudo-code for determining the keyword group order and keyword-order pairs

After all the terms have been scanned, the term-by-group matrix is computed for the documents. In term-by-group matrix, rows correspond to terms in the document; columns correspond to keyword group and cells correspond to frequency of term occurrence in the keyword group. Table 4 shows the term-by-group matrix for document vector *D1*. The types of group are denoted as Type 1 for standalone keywords, Type 2 for group with two keywords, Type 3 for group with three keywords.

Table 4: Term-by-group matrix for D1

Term	Non Keyword-order Group			Keyword-order Group	
	Type 1	Type 2	Type 3	Type 2	Type 3
shipment	0	0	0	0	1
gold	0	0	0	0	1
silver	0	0	0	0	1
damage	1	0	0	0	0
fire	1	0	0	0	0

6.0 TERM WEIGHT ASSIGNMENT

The proposed weight function $F(n,g,k)$, is defined as an exponentially increasing function based on group size, group order and the number of keyword-order pairs found in the group. It can be written as shown in Formula (1).

$$F(n,g,k) = \begin{cases} 2^{n-1}, \text{ when } g = 0 \text{ (for non keyword-order group)} \\ k\,(2^{n-1}), \text{ when } g = 1 \text{ (for keyword-order group) and,} \\ \qquad k = \textit{Number of keyword-order pairs} \end{cases} \quad (1)$$

Thus, for a document vector D_i expressed in $\{w_{i,1}, w_{,i,2}, \ldots, w_{i,M}\}$, the new term weight $x_{j,}$ corresponding to element $w_{i,j}$ of vector D_i is then computed as the summation of weight in all corresponding group type as shown in Formula (2).

$$x_j = \sum_{n=1}^{N} w_{j,n}\, F(n,0) + \sum_{n=2}^{N} w_{j,n} * F(n,1)$$

$$= \sum_{n=1}^{N} w_{j,n} * 2^{n-1} + \sum_{n=2}^{N} w_{j,n} * k * 2^{n-1} \quad (2)$$

Where,
n = size of the group
k = number of keyword-order pairs
$w_{j,n}$ = frequency of term occurrence found in the group

For query vector Q expressed in $\{w_{q,1}, w_{q,2}, \ldots, w_{q,M}\}$, the query weight $w_{q,j,}$ corresponding to each term j is given value of 1 when term j is a keyword or zero when term j is a non-keyword.

The resulting document vector *D1* and query vector is as presented in Table 5.

Table 5: Document and query vector representation for D1

Term	Group 0			Group 1		Term Weight x_j	Query Weight $w_{q,j}$
	1	2	3	2	3		
shipment	0	0	0	0	1	8	1
gold	0	0	0	0	1	8	1
silver	0	0	0	0	1	8	1
damage	1	0	0	0	0	1	0
fire	1	0	0	0	0	1	0

7.0 SIMILARITY MEASURE

The similarity between document and query vectors can be calculated using cosine measure, [8], as shown in Formula (3).

$$sim(Q, D_i) = \frac{Q \cdot D_i}{|Q|\,|D_i|} \qquad (3)$$

$$= \frac{\sum_{j=1}^{M} x_{i,j} * w_{q,j}}{\sqrt{\sum_{j=1}^{M} x^2_{i,j}} * \sqrt{\sum_{j=1}^{M} w^2_{q,j}}}$$

8.0 SYSTEM PERFORMANCE EVALUATION

The effectiveness of the proposed model was evaluated using precision metric, which measures the number of relevant retrieved results. It is the proportion of retrieved documents that are relevant and is defined as Formula (4) below:

$$\text{Precision} = \frac{\text{Number of relevant retrieved}}{\text{Total number retrieved}} \qquad (4)$$

The results of the proposed MTAKO model was compared with the classic vector space model (Classic VSM) and Multi-Term Vector Space Model (MTVSM), both of which are not based on keyword-order relationship. The proposed model was also measured in terms of the percentage of average precision improvement of the Top-20 retrieved results.

9.0 EXPERIMENT SETUP

Sample data from Communications of the ACM (CACM) collection was setup for the experiment. The search is based on the titles and abstracts of articles from the collection. A set of 15 randomly picked queries from the collection was manually converted to keyword-based queries to suit the proposed keyword-based search environment.

10.0 RESULTS

The experiments were performed on three different ranking models: MTAKO, MTVSM and classic VSM. The comparison of the system performances will include the query-by-query average precision, precision histogram, precision at cut-off rank and mean average precision.

Table 6 shows the query-by-query average precision analysis depicting the number of queries for which MTAKO achieved higher average precision (+), lower average precision (-) or same average precision (=) for a query.

Table 6: Query-by-query average precision analysis

Models Comparison	Precision Differences			Total Queries
	+	-	=	
MTAKO – MTVSM	12	3	0	15
MTAKO – Classic VSM	11	4	0	15

Table 6 shows that MTAKO achieved 12/15 queries with average precision higher than MTVSM and achieved 11/15 queries with average precision higher than classic VSM. The higher average precision values indicate that more of the highly ranked documents were relevant, [[9], clearly showing that MTAKO model outperformed both the MTVSM and classic VSM models for the Top-20 retrieved documents.

The average precision obtained for each of the queries is further analyzed for its difference in percentage using precision histogram. Let M_A be the system with model A and M_B be the system with model B, if the average precision percentage difference is zero, this indicates that M_A has the same level of performance with M_B for a query. A positive percentage difference (> 0%) implies that performance of M_A is better while a negative percentage difference (< 0%) implies that the performance of M_A is worse than M_B.

Fig 2(a) shows the performance comparison between MTAKO and MTVSM while Fig 2(b) shows the comparison between MTAKO and classic VSM models.

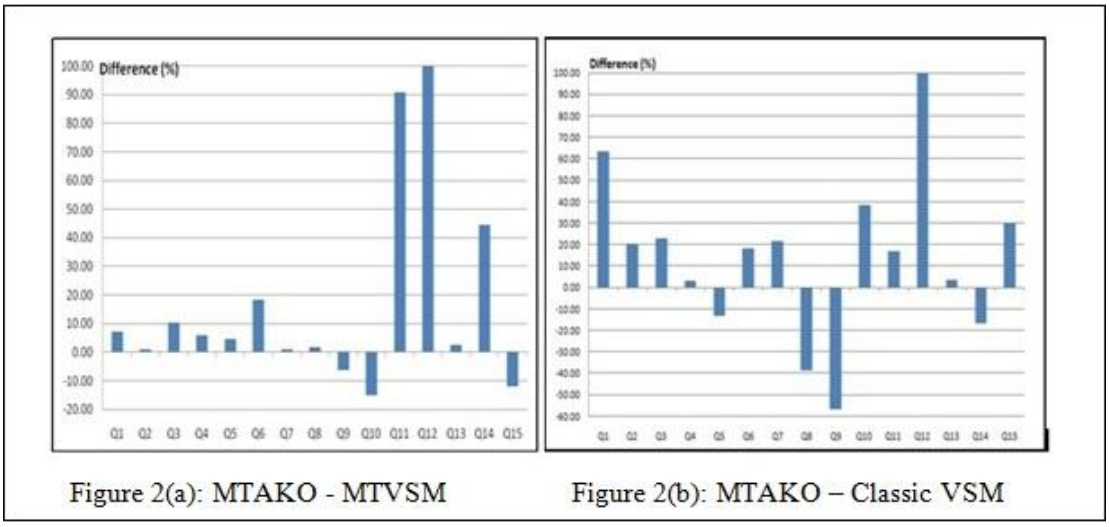

Fig. 2. Precision histogram for average precision performance comparison

Performance comparison between MTAKO – MTVSM revealed better performance for 12 queries (Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q11, Q12, Q13, Q14) and worse performance for 3 queries (Q9, Q10, Q15). The higher average precision differences for the 12 queries demonstrate that MTAKO is more effective, and thus performs better than MTVSM. Similarly, the performance comparison between MTAKO – classic VSM shows that MTAKO revealed better performance for 11 queries (Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q11, Q12, Q13, Q14) and worse performance for only 4 queries (Q9, Q10, Q15), indicating that MTAKO performs better than classic VSM model.

The average precision at the cut-off rank of Top-5, Top-10, Top-15 and Top-20 retrieved documents for MTAKO, MTVSM and classic VSM are shown in Table 7.

Table 7: Average precision at cut-off rank

Average Precision			
Cut-off rank	MTAKO	MTVSM	Classic VSM
5	0.4000	0.4000	0.4400
10	0.3733	0.3467	0.3267
15	0.3111	0.3067	0.2978
20	0.2533	0.2500	0.2700

The results show that at cut-off rank of Top-5 and Top-20 retrieved documents, classic VSM outperforms both the MTAKO and MTVSM models. However, for the majority of the retrieved documents (from 6th to 15th retrieved documents) at cut-off rank of Top-10 and Top-15 , MTAKO shows higher average precision values than MTVSM and classic VSM, indicating that more relevant documents are retrieved in this range. This shows that MTAKO model performs better than MTVSM and classic VSM.

Finally, the mean average precision provides another way of comparing the system performance that reflects the performance over all the relevant documents. Table 8 shows the mean average precision values achieved by MTAKO (56.97%), MTVSM (50.14%) and classic VSM (55.55%) over the 15 queries, clearly indicating MTAKO performs the best among the three models.

Table 8: Mean average precision

Model	Mean average precision (%)
MTAKO	56.97%
MTVSM	50.14%
Classic VSM	55.55%

The MTAKO model gives an average precision improvement of 6.83% over MTVSM and 1.42% over Classic VSM for the Top-20 retrieved documents. Again, this proves that the overall performance of MTAKO model is superior compared to MTVSM and classic VSM.

9.0 CONCLUSION AND FUTURE WORK

This study introduced an enhanced relevance ranking technique for information retrieval based on vector space model. The proposed model focuses only on precision that favors to system which retrieves relevant documents quickly, that is early in the ranking. Classic vector space model and Multi-Term Vector Space Model work well in retrieving most of the relevant documents but, with additional consideration of terms adjacency in keyword-order relationship in Multi-Term Adjacency Keywords Order (MTAKO) model tends to improve the precision in the overall retrieved results. Some of the future research to enhance MTAKO is to include indexing approach that may improve the system efficiency. The current study performed the document indexing during runtime in which the indexes were not pre-processed in the collections. This may incur high processing cost in a very large text collection. Future work may also explore on efficiency of indexing method in the system. The semantic technology such as the study of the meaning of terms or words of the query may be explored to further increase the retrieval effectiveness of the search results as discussed in [[10]. Other further research would be to incorporate the relevance feedback technique (a technique to reformulate the original query based on the feedback from the initial query results), [[10]. As presented in [10], the use of a VSM based scheme is very useful in even natural language processing (NLP). If our model were combined with that shown in [10], it could potentially give MTAKO a new avenue of usefulness in the area of NLP.

REFERENCES

[1] A. M. Z. Bidoki, P. Ghodsnia, N. Yazdani and F. Oroumchian, "A3CRank: An adaptive ranking method based on connectivity, content and click-through data", *Information Processing and Management*, 2010, Vol. 46: pp. 159–169.

[2] G. Salton, A. Wong and C. S. Yang, "A vector space model for automatic indexing", *Communications of the ACM*, November 1975, Vol. 18(11), pp. 613-620.

[3] X. Tai, F. Ren and K. Kita, An Information Retrieval Model based on Vector Space Method by Supervised Learning, *Information Processing and Management*, 2002, Vol. 38, pp. 749–764.

[4] I. R. Silva, J. N. Souza and K. S. Santos, "Dependence Among Terms in Vector Space Model", *Proceedings of the International Database Engineering and Applications Symposium* (IDEAS'04), 2004.

[5] L.S. Wang, "*Relevance Weighting of Multi-Term Queries for Vector Space Model*", IEEE, 2009.

[6] L.S. Wang and D.X. Wang, "Method for ranking and sorting electronic documents in a search result list based on relevance", *U.S.Patent Application Publication US 2007/0179930 A1*, January 31, 2006.

[7] D. L. Lee, H. Chuang and K. Seamons, "Document Ranking and the Vector Space Model", *Software, IEEE*, 1997, Vol. 14(2): pp. 67-75.

[8] G. Salton, E. A. Fox and H. Wu, "Extended Boolean Information Retrieval", *Communications of the ACM*, December 1983, Vol. 26(12).

[9] R.B. Yates, and B.R. Neto, "*Modern Information Retrieval*", Addison Wesley Longman, 1999.

[10] R.G. Raj and V. Balakrishnan, "A Model for Determining the Degree of Contradictions in Information". *Malaysian Journal of Computer Science*, September 2011, Vol. 24(3): pp. 160-167.

[11] G. Salton and C. Buckley, "Improving Retrieval Performance by Relevance Feedback", *Journal of the American Society for Information Science*, 1997, Vol. 41(4), pp. 288-297.

BIOGRAPHY

Lim Bee Huang is a student in University of Malaya (UM) attached to the Department of Information Systems. She has just completed her Masters studies in the field of information retrieval.

Vimala Balakrishnan holds a PhD from Multimedia University (MMU). She is currently a senior lecturer at the Department of Information System of the Faculty of Computer Science and Information Technology at UM. Her work deals with information security, information retrieval and data and knowledge engineering.

Ram Gopal Raj holds a PhD from the University of Malaya (UM) and is attached with the Department of Artificial Intelligence of the Faculty of Computer Science and Information Technology at UM. His research is mostly related to chatterbot technology and information search methods as well as knowledge representation schemes.