VOL. 10, NO. 3, FEBRUARY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

COMPARATIVE WEIGHTING METHODS OF VECTOR SPACE MODEL

Sasithorn Lertariyatham[1], Pongpisit Wuttidittachotti[1], Somchai Prakancharoen[2], and Sakda Arj-ong Vallipakorn[3]
[1]Faculty of Information Technology, King Mongkut's University of Technology, North Bangkok, Thailand
[2]Faculty of Applied Science, King Mongkut's University of Technology, North Bangkok, Thailand
[3]Faculty of Medicine, Ramathibodi Hospital, Mahidol University, Bangkok, Thailand
E-Mail: mali3fja@gmail.com,

ABSTRACT

This research aimed to develop a program for data retrieval stored in the form of questions and answers. Fidility Fascinate Fastness Co., Ltd., Thailand has been used an old traditional of storage and retrieval of knowledge system in the form of Google Drive, which was inconvenient and time consuming when retrieving the desired knowledge. Therefore, the new development of knowledge retrieval based on Vector Space Model (VSM) to facilitate the users in the knowledge retrieval was conducted and invented to solve the problems. For VSM concept, the required knowledge from the database was transformed by with C# and wrapped by the Longtext Matching, then indexed cutting by Inverted Indexing Search. Information retrieval and sorting results was robustness based on algorithm of VSM. The results of knowledge retrieval of 200 questions were processed by 100 queries. The Cosine formula shows the best appropriated formula than Dice and Jaccard formulas which return the higher of their precision (82.50 %), recall values (97.35%), and accuracy (89.31%) measured by F-measurement.

Keywords: long text matching, information retrieval, vector space model, VMS.

INTRODUCTION

The Fidility Fascinate Fastness Co., Ltd., Thailand has an old system for storage and retrieval of knowledge in the form of Google Drive, which was unpractical and difficult to use when retrieving required knowledge. The new conceptual model of knowledge retrieval aims to store and retrieve information in form of questions and answers in the C # language. Then, it was wrapped using long text matching, indexed prompt could be retrieved any word lists in the question and established to the users. The system would display the questions with the groups of example sentences with the word retrieval appearing with linked to the answers. From the operation of the system, it also provided the ability to record key words, predefined questions and answers. In this way, the returned results will be shown in both matched question and answer, meeting the needed of users, allowing for easily and rapidly retrieved and more effectiveness.

Question-answering system

Sansarn Look, one type of information retrieval system (Information Retrieval) was a request (Query) in the form of natural language, and gained accurate and fast answer results (Haruechaiyasak C., 1994). The process of data access methods of information retrieval system selected and retrieved only information that the user needs. In addition, the information filtering function was integrated to get rid of the irrelevant unwanted information. The main development concept was focuses on the comparative importance of the documents, and term from users (Keywords Matching). The components of this information retrieval system were divided into three designed stages.

Stage-1: To create the index (Indexing) associated with the knowledge system.

Stage-2: To search from the index (Retrieval) related to the user requirement

Stage-3: Select the algorithm to compare the results from the searching of index (Retrieval) to finalize the best one of efficient algorithm model.

Longtext matching

It was the longest word which found in the dictionary (Welukaman, T. and Prakancharoen, S. , 2 0 1 , the word "person" as a noun when used with verbs such as "the", "sign up" to the word "representative" "the candidate" a word whose meaning in dictionary. Example of the sentences, "The prosecution case was appealed through the hotline Bangkok" is wrapping. In to “the law-suit - have - health care - through - Hotline – Bangkok” etc.

Inverted indexing

Inverted Indexing was a popular method in information retrieval systems (Salton, G., Anita Wong and Chung-Shu Yang, 1975). Because of its high efficiency in terms of speed and space-saving storage consisted of two parts. Examples of Inverted Indexing structure were shown in Figure-1: Documents 1, labeled as "The time-stamping machine is out of order", and Documents 2 labeled as "The company require everyone to sign in on their working time".

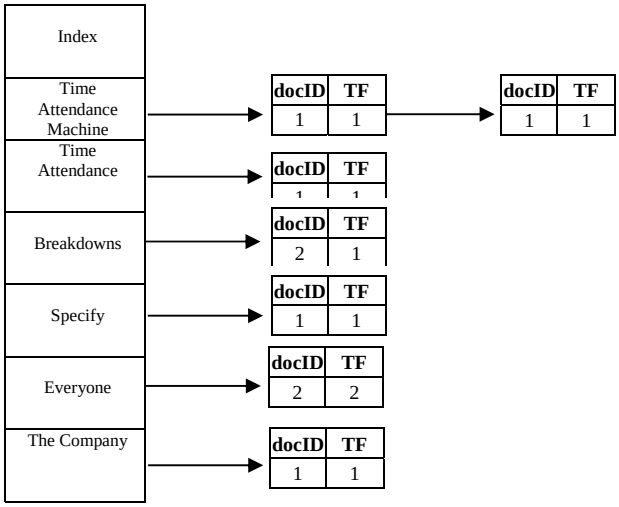


Figure-1. Examples of inverted indexing structure.

Table-1. Example of indexed storages, data shown in the form of sets.

Word	Documents found
Time Attendance Machine	{1,2}
Time Attendance	{1}
Breakdowns	{1}
Specify	{1}
Everyone	{1}
The Company	{1}

In Table-1 , the retrieval of the sentences "The time-stamping machine was broken down" when using the algorithm to cut the word, it transformed in to two words "time-stamping machine", "Out of order" after retrieval. That mean inverted index set was {1, 2} {1} = {1}

Vector space model

We put the words into a document then arranged them into a vector format, each vector represented the words in each documents (NAZT Apache Software Foundation, 2010). They compared the similarities of each document by measuring the angle between the vectors axis using the cosine or dot product to measure the angle of differences. From the results, the lower value showed the similarities of results.

The cosine value was between 0 and 1. If the cosine was close to 0, it showed no similarity at all. But, if the cosine value was close to 1, it means that the document was very similar, as shown in Figure-2-4.

Assigned "X" as the horizontal main axis

Assigned "Y" as the main vertical axis.

Doc 1 assigned as the words in a document 1 that has been sorted into a vector format.

Doc 2 assigned as the words in a document 2 that has been sorted into a vector format.

"q" represented the query terms of user retrieval.

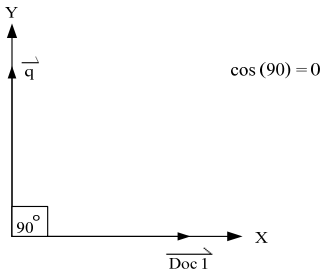


Figure-2. The query with no similarities.

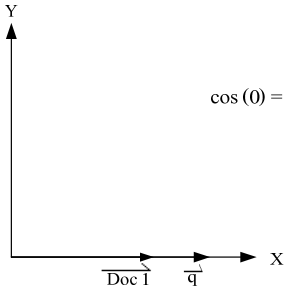


Figure-3. The query documents are most similar.

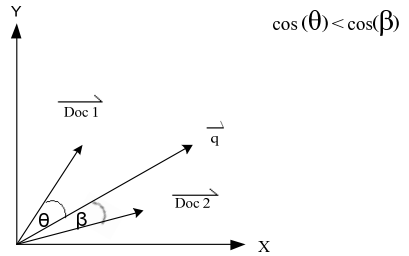


Figure-4. The query was similar to the Doc 1 less than the Doc2.

This document retrieval system will return the results with high to low similarity scores between correlation of queries and documents of cosine or dot-product of vector of Vector Space Model.

M was assigned as the total inverted indexes of system.

N was the total documents in the system.

J was the order of words of retrieved documents

I was the order of inverted indexes in retrieved documents.

$w_{i,j}$ was the weight of inverted indexes.

tf was the frequency of retrieval term.

idf was the frequency of retrieval term showed in documents.

$f_{i,j}$ is the frequency of term ti in j document.

n_i is the number of total documents which represent term i.

log was log 2 or natural logarithm e or ln

$Max_k f_{k,j}$ was maximum of frequency show in document j.

Thus, the ordering of the document from 1 to N will assign as Doc and write in the matrix system as.

$$Doc = [Doc_1 \ Doc_2 \ \dots \ Doc_j \ \dots \ Doc_n]$$

Table-2. The arrangement by order of weighted of words in the documents.

	word1	word2	word3	…	wordn
doc1	weight1,1	weight1,2	weight1,3	…	weight1,n
doc2	weight2,1	weight2,2	weight2,3	…	weight2,n
…	…	…	…	…	…
docN	weightN,1	weightN,2	weightN,3	…	weightN,n

From Table-2, the results showed the weight of word in the documents and transform in to vector format as below;

Vector of document 1;

{ w1,1 , w1,2 , w1,3, … , w1,2n}

Vector of document 2;

{ w2,1 , w2,2 , w2,3, … , w2,2n}

Vector of document N

{ wN,1 , wN,2 , wn,3, … , wn,n}

Then, Docj will wrote in to;

$$Doc_j = [w_{1,j}\ w_{2,j} \ldots w_{i,j} \ldots w_{M,j}]$$

$$w_{i,j} = tf_{i,j} \times idf_i$$

$$tf_{i,j} = f_{i,j} / (Max_k f_{k,j})$$

$$idf_i = \log (N / n_i)$$

Similarity calculation = similarity score $Sim(q, d_j)$

$$\sum_{i=1}^{M} w_{i,q} \times w_{i,j}$$

The total of frequency of query * weight of term

$$\sqrt{\sum_{i=1}^{M} w_{i,q}^2}$$

Square root the total sum up of weights from queries power 2.

$$\sqrt{\sum_{i=1}^{M} w_{i,j}^2}$$

Square root the total sum up of weights from queries power 2

The similarity calculation

$$Sim(q, d_j) = \frac{\sum_{i=1}^{M} w_{i,q} \times w_{i,j}}{\sqrt{\sum_{i=1}^{M} w_{i,q}^2} \times \sqrt{\sum_{i=1}^{M} w_{i,j}^2}} = \cos\theta$$

In case of q show the same direction as d_j will show cosine = 1, that mean maximum of matched.

If q made an angle of 60 degree d_j will earn cosine = 0.5.

If q made an angle of 90 degree d_j will earn cosine =0 or completely unmatched.

For sorting document based on similarity properties from 1 to N and the results will use to compare by d_k important ranking than sequences of d_j e.g. Sim (q, d_k) >Sim (q, d_j).

Study and analysis system

In general practices, many companies usually store their knowledge data on google drive, these causes inefficiency of usages when searching and retrieving information. Most problems referred to wrong answers or irrelevant answers, which appeared to unappropriated match requirements from searching. This research aimed to develop an algorithm to cut long sentences of knowledge data via long text matching. Results are then wrapped once by C# methodology of question-answer searching and retrieval. Then transformation of this cut words into indexes similar with human language. The users can be retrieved any of the word-listed questions from the keyword that entered into the system.

For example, the question "Who was the first prime minister of Canada" the system will try to understand, cut and brought this word to compare with indexing words in the system. That mean the users can search and retrieve any words in the sets of question and answer. This showed with pairs of questions and answers with word retrieval appearing with examples of relevant sentences.

The result of the outcome showed in calculated percentage that allow the user to decide how to show interrogatives displays. The results also linked to answers by the operation of the system, which was allowed users to record key words. To create Predefined Question and Answer –this method returned data questions with an answer to meet user needs with rapidly response and high efficiency.

System design

The designed system used to create an index (Indexing Word) and information retrieval algorithm (Information Retrieval) is shown in Figure-5 and schematic operation Figure-6.

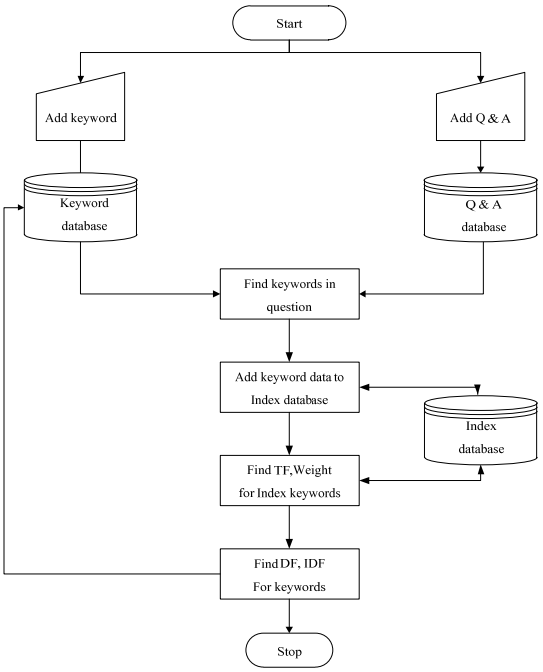


Figure-5. Diagram shows the working processes.

The system to create a database index (Indexing Word) worked by importing data and cutting words by the LexTo, National Electronics and Computer Technology

Center (NECTEC), Thailand. The author of this paper entered 200 sentences of knowledge from company employee to create pilot system database. The system made two image databases cut by long words and kept, provided the meaning and storing as much as possible by the Longtext Matching technique. The system calculated the TF, Weight to Database Index (Indexing Word) to estimate the similarity df and IDF of each term from the $w_{i,j} = tf_{i,j} \times idf_i$.

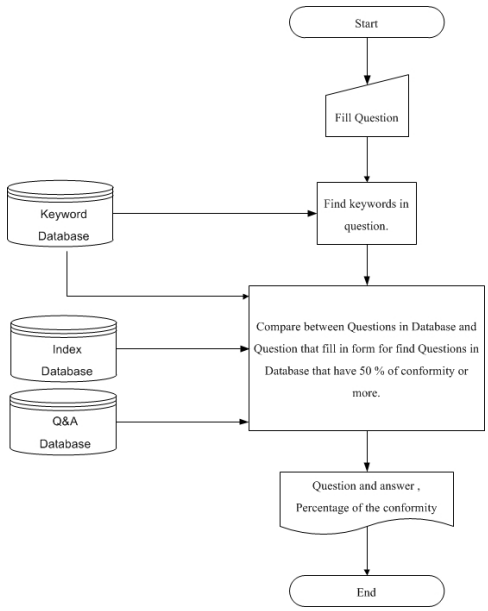


Figure-6. Diagram shows the conceptual workflow processes of the Information Retrieval System.

The Information Retrieval System by the VSM was developed using the Cosine formula to calculate the accuracy (precision), recall values, and validity (F-measure) from three formulas, which included Cosine, Dice, and Jaccard. From the analyses, overall three formulas, Cosine formula seemed to have a good result of average accuracy (Precision), Recall and the validity (F-measure) than the remaining two formulas. The authors summarized that the Cosine formula was most appropriated to develop the system. Because of its accuracy, efficiency and ability to showed questions and relevant answers to users.

The formula

Cosine
$$Sim(D,Q) = \frac{\sum_i (a_i * b_i)}{\sqrt{\sum_i a_i^2 * \sum_i b_i^2}}$$

Dice
$$Sim(D,Q) = \frac{2\sum_i (a_i * b_i)}{\sum_i a_i^2 + \sum_i b_i^2}$$

Jaccard
$$Sim(D,Q) = \frac{\sum_i (a_i * b_i)}{\sum_i a_i^2 + \sum_i b_i^2 - \sum_i (a_i * b_i)}$$

The steps of work, began with user input a question that they needed an answer into the system using a wrapping by longest text matching technique (Longtext Matching). The cutting words have to calculate the fitness value with Cosine, Dice and Jaccard formulas, then further analyses by compared these cut word of question with indexing of question in storage database and selecting the most matched and similar results more than 50 % showed to user. The users can be selected the results from the pool of 50-100 % in which relevant questions and find out the answers that they needed. The system will show the relevant percentages with the search results to guided users to decision making.

Development

System was developed under web application in C # language using the Windows 7 operating system, MSSQL database management system. For web development, Adobe Dreamweaver CS4 was used and designed user-friendly web pages with Adobe Photoshop CS4 system. The Data Flow Diagrams (DFDs) were used as a tool to analyze the system, in order to understand how the system worked in relation to information. DFDs showed the flow of imports and exports of data, which were divided into different levels. The Data Dictionary (Data Dictionary) was created and help to explain the structure of the tables. These help the users to understand the database in meaning of each entity and relationships, including types and sizes. The main system database was divided into 4 database parts which consisted of administrator database, Keyword database, Indexing word database, and question and answer databases with wrapping techniques with the longest truncation Long text Matching.

The main structure of system consisted of two domain separated parts on prospective views: 1) the administrator part, and 2) the user part.

Testing

The testing process was done by storing and retrieving knowledge with VSM. The 1 0 0 queries of test questions and the results of 200 questions using Cosine, Dice and Jaccard testing formulas were collected.

Searching and indexing

These developments of storage and retrieval of knowledge with the VSM is practically divided users into the part of the administrator and the end user part. The index are created from words in the Lento program of the National Electronics and Computer Technology Center, Thailand. Enters of questions and answers from knowledge of the personnel in the company. The system also provides the Advanced Boolean searching function with all of the words (AND) of a sentence or phrase, or at least one word. (OR), and no words found at all. (NOT) using for default searching and Wildcard. The results display as a question and shows the similarity at a rate of at least 50 percent, and the users can also add words, questions, and answers into the system in order to increase the useful amount of knowledge storages in the system, as shown in Figure-7-12.

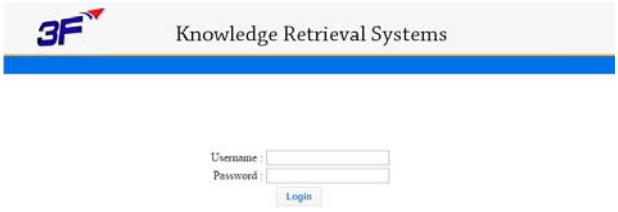


Figure-7. The login screen of system.

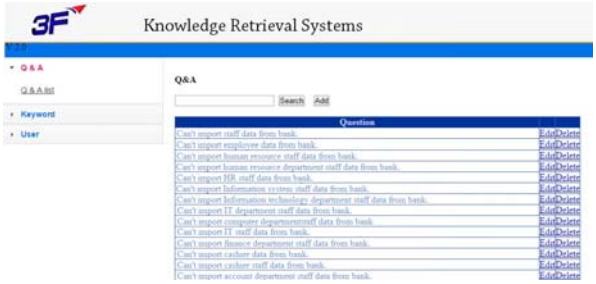


Figure-8 shows the questions entries into the system.

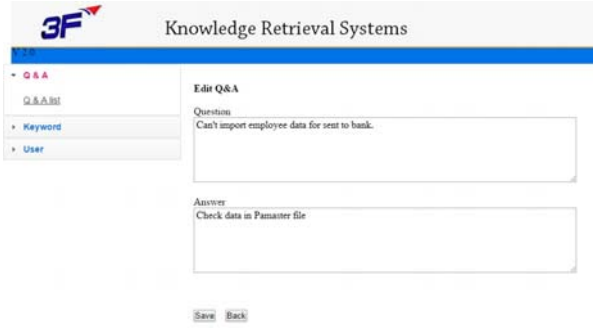


Figure-9. Resolved questions.

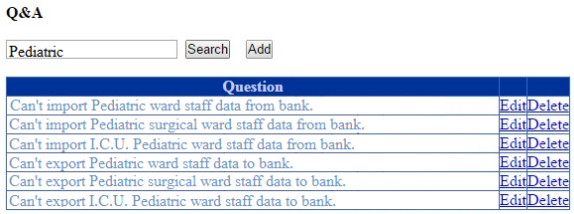


Figure-10. Searching results.

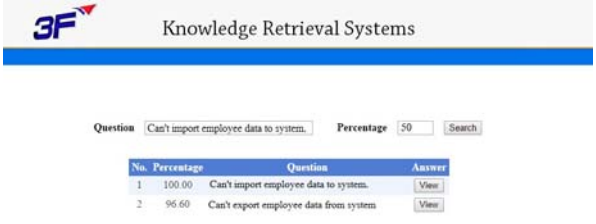


Figure-11. The most matched results of questions.

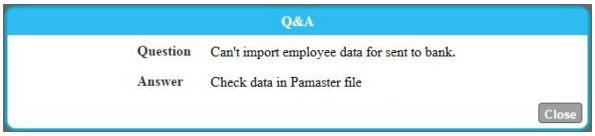


Figure-12. Pop-up shows the questions, answers and recommendation.

The measured system performance

We measure efficiency of the system by testing the three formulas including Cosine, Dice, and Jaccard formula as follows.

Cosine found that average of fitness values equal to 0.4990946. Dice found that the average of fitness values equal to 0.3946528, and Jaccard found that the average of fitness values equal to 0.24229466.

DISCUSSIONS

The system was designed and developed with a sequence of operations as follows: First, brief information regarding the operating system needs to study the issue was creates and constructed a conceptual framework by UML (Unified Modelling language). The second step was overall analyzing of the systems design using Data Flow Diagram (DFD). The final step was system testing and verifying the correctness of a program or efficiency of work. The concepts of development storage and retrieval of knowledge with the VSM of the system can be summarized as follows.

1) To find out the question - answer to meet the conditions and needs.

2) Keywords management.

3) Users decision process.

4) Performance test by using 100 Queries and 200 questions. The questions for key words and testing by 3 formulas Cosine, Dice and Jaccard to calculate the fitness values.

5) The overall evaluation of the usages into 3 aspects found that Cosine formula had a highest average value of 0.4990946 for the fitness value which most appropriate to use and implement to this designed system.

Routing problems

The model used to design and development of storage and retrieval of knowledge of the process model is the vector space mode require well understand and knowledge of personnel, high level of working experience to save words or cognitive in nature. This is an important of user performances to answer questions in order to get the knowledge to be effective, helpful to users, recording and save the new knowledge into the system without redundant information. These is essential component to provide higher of efficiency and benefits of the system.

CONCLUSIONS

Development of knowledge retrieval from a database requires C # and wrapping by wrapping the longest (Long text Matching), then cutting words are indexed by Inverted Indexing, retrieval and sorting the

results. The information retrieval by VSM has more efficiency by using Cosine formula to calculate the accuracy (Precision), Recall and the validity (F-measure). The experimental knowledge retrieval system used 100 queries of 200 questions entries to test the accuracy of the documents earn more benefits and efficiencies, but the completeness of the documents show in the moderate rating.

Recommendations and guidelines for further development

The accuracy of the results can be increased by adding the relevant and correct words in the dictionary of the system by capturing it from new knowledge from the organization. In another way, the more accurate will come with contrary from increasing vocabulary and excessive burden to loading of system memory. This mean the developer or the user needs to be carefully done and should enhance their ability to find out of answers with correct meaning keyword.

REFERENCES

Bangcharoensup, P. and Jaikaew C. (2009-2010). A Machine-Translation based Approach to Word Boundary Identification: A Projective Analogy of Bilingual Translation, Bangkok: NECTEC BEST2010 (Benchmark for Enhancing the Standard of Thai language processing)

Chanta, S. and Porrawatpreyakorn, N. A Web News Information Classification and Retrieval System using Multilayer Perceptron Neural Network. Information Technology Journal, Vol.9, No.2, (July-December), 2013.

Deborah L. McGuinness and Frank van Harmelen. (2004).

Özgür Öztürk. Feature Extraction and Similarity-Based Analysis For Proteome And Genome Databases. Bachelor Thesis, Ohio State University, 2007.

Dumais, S. et al. Web Question Answering: Is More Always Better? Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere. 11-15 August 2002.

Fergus, P. et al. Capturing Tacit Knowledge in P2P Networks. Proceedings of 4th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, Liverpool. 16-17 June 2003.

Haruechaiyasak C.(1994). Developing IR System via Lucene. National Electronics and Computer Technology Center (NECTEC).

Innerhofer, R. Using Approximate String Matching Techniques to Join Street Names of Residential Addresses. Bachelor Thesis, Faculty of Computer Science, Free University of Bolzano, 2004.

Matsumura, N., Ohsawa, Y. and Ishizuka, M. (2001). Knowledge Navigation on Visualizing Complementary Documents. Lecture Notes in Computer Science, 2226, pp. 258-270.

Meesuj, P. Nuipean W and Bunrod, P. Semantic Search for Information System Domain Bibliographic Data. Journal of Information Science and Technology. 2013, pp.11-20.

Michael K. Brown. Stochastic Language Models (N-Gram) Specification, W3C, 2001.

Moldovan, D. et al. LASSO: A tool for surfing the answer net. Proceedings of the Eighth Text Retrieval Conference, Maryland. 17-19 November 1999.

Paranan, M. and Jeerungsuwan, N. Development and Efficiency of TheWebApplication for TQF. Courses Specifications Making Support via Information Retrieval: VectorSpace Model Technique. Technical Education journal King Mongkut'sUniversity of Technology North Bangkok. 2014.

Pasca, M. A., and Sanda M. Harabagiu. High performance question/answering. Proceedings of the 24th international Conference on Research and Development in Information Retrieval, Louisiana, 9-13 September 2001.

Phiakoksong, S and Chamnongsri, N. (2010). A Knowledge Navigatio System for Accessing Contents in Prited Materials. Journal of Information Science, Vol. 28, No.3,(September-December), 2010.

Qiu, J., Yao, Y., Wang, Y. and Wang, X. (2006). Research of E-Government Knowledge Navigation System Based on XTM. Proceedings of the 2006 IEEE/WIC/ ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IATW'06).

R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval, ACM Press, Addison Wesley, 1999.

Salton, G., and Christopher Buckley. Term-weighting approaches in automatic text retrieval. Information Processing & Management 24, 5 (1988), pp.513-523.

Subramaniam, S. et al. Knowledge engineering for protein structure and motifs: design of a prototype system. Fourth International Conference of Software Engineering and Knowledge Engineering, Capri. 15-20 June 1992.

Salton, G., Anita Wong and Chung-Shu Yang. A vector space model for automatic indexing. Communications of the ACM 18, 11 (November 1975), pp. 613–620.

Tao, X. & Li, Y. (2009). Concept-based, personalized web information gathering: A survey. In proceedings of the 3rd international conference on knowledge science, engineering and management. KSEM'09, November 2009, pp. 21-228.

Thada, V. and Jaglan, V. Comparison of Jaccard, Dice, Cosine Similarity Coefficient To Find Best Fitness Value for Web Retrieved Documents Using Genetic Algorithm. International Journal of Innovations in Engineering and Technology. 2013, pp.202-205.

Usanavasin, S. Non-Dictionary-Based Thai Word Segmentation Using Decision Trees, Sinrindhorn International Institute of Technology, 2003.

Vallet, D. et. al. (2007). Personalized content retrieval in context using ontological knowledge. IEEE transactions on circuits and systems for video technology. 17(3): 336-346.

Voorhees, E. and Dawn M. Tice. The TREC-8 Question Answering Track Evaluation. Proceedings of the Eighth Text Retrieval Conference, Maryland. 17-19 November 1999.

W. B. Frakes and R. Baeza-Yates, eds., Information Retrievall: Data Structures & Algorithms, Pretice Hall, 1992.

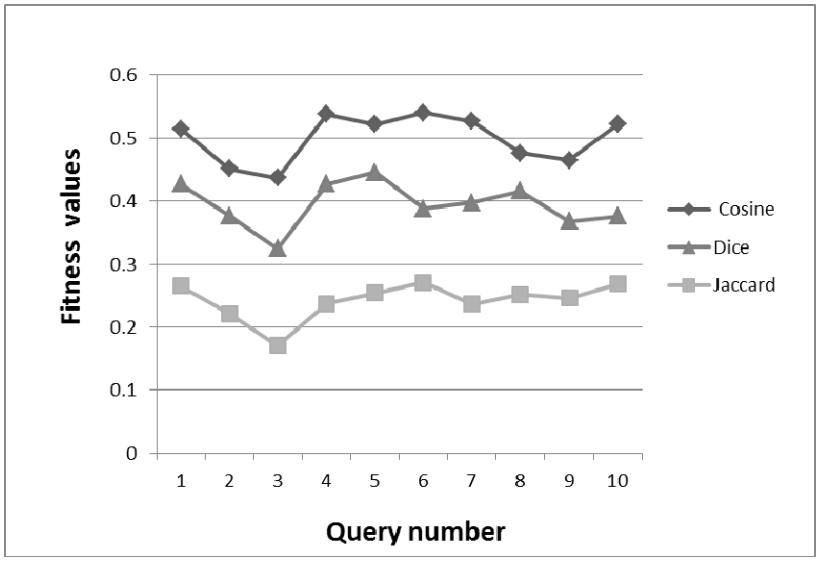
Welukaman, T. and Prakancharoen, S. (2011). A Case Study of Administrative Court the 23rd National Graduate research Conference, pp.190-196.

Wuttikriengkraipol, A. (2011). Optimization to Information electronics file retrieval system, by Fuzzy logic. Special problem of Science , Master of Science in Information Technology, Faculty of Information Technology, King Mongkut's University of Technology North Bangkok.

APPENDIX

Table shows the best fitness values among different queries by Cosine, Jaccard and Dice formulas

No.	Question	Cosine	Dice	Jaccard
1	Cannot log in to canteen system by identity card	0.514726	0.426824	0.265556
2	Unavailable canteen pos machine	0.451263	0.376805	0.221118
3	Clock in button of time attendance machine	0.436438	0.324856	0.170834
4	Cannot Import employee data to system	0.538142	0.426879	0.23726
5	Therapeutic radiology and Oncology department staff	0.521418	0.445895	0.254336
6	Orthopaedics Operating Room staff data	0.540459	0.387621	0.271144
7	Special male-female surgical ward staff data	0.526422	0.397308	0.236416
8	Cannot use card	0.475673	0.416221	0.251868
9	Canteen management system with RFID card	0.464643	0.367892	0.2456856
10	Import cashier data for sent to bank	0.521762	0.376227	0.268729



Comparison of similarity coefficients for fitness value among 3 formulas

Benchmarking of three formulas among Cosine, Dice, and Jaccard as follows;

Cosine test was an average of fitness values equal to 0.4990946.
Dice test was an average of fitness values equal to 0.3946528.
Jaccard test was an average of fitness values equal to 0.24229466.

The overall estimation of the testing of storage and retrieval of knowledge with the VSM
The Cosine formula showed a highest accurate and precision of information retrieval.