A FRAMEWORK OF QUESTION ANSWERING SYSTEMS FOR DIABETES CARE USING LATENT SEMANTIC INDEXING WITH TEXT MINING

Katsara Phetkrachang[1] and Nichnan Kittiphattanabawon[2]

[1]Doctor of Philosophy Program in Management of Information Technloogy, School of Informatics, Walailak University, 222 Thaiburi district, Thasala, Nakhon Si Thammarat 80161, Thailand, ketsara.p@rmutsv.ac.th

[2]Lecturer, Management of Information Technloogy, School of Informatics, Walailak University, 222 Thaiburi district, Thasala, Nakhon Si Thammarat 80161, Thailand, knichcha@wu.ac.th

ABSTRACT

Currently, question answering systems still have some problems due to the ambiguity of words. Sometimes, the words with the same meaning, but differently writing can bring the wrong answers. Latent Semantic Indexing (LSI) is one method that many researchers used to solve a problem of synonym since LSI can be applied for finding the latent semantic of the synonym. Moreover, LSI also reduces the document size while their meaning remains. This paper presents a conceptual framework for the development of a question answering system using LSI. Here we applied the question answering system for diabetes care. The framework consists of three main steps, i.e., (1) document pre-processing, which is applied by a technique of text mining, (2) LSI answer scoring, which follows LSI methods by term frequency-inverse document frequency (TF-IDF) weighting, and (3) question answering matching, which use the similarity measurement. This paper also includes examples of each step. A preliminary experiment shows that the conceptual framework offered can provide the correct answer.

KEYWORDS: LSI, text mining, question answering system, diabetes, TF-IDF

1. Introduction

A question answer system still has problems on finding an efficient answer since the system still mostly used keywords to match a similarity of words in a question and those of which in an answer collection. As some keywords are synonym, so these would lead mostly to wrong answer. LSI is a technique that supports in finding a semantic meaning of word that is written differently but have the same meaning. There are several researches which showed that LSI is effective for alleviation of these problems. Al-Anzi et al. [1] proposed a measurement of the Arabic language text classification comparing between vector space model (VSM) and LSI using with Naive Bayes and K-NN. The results showed that LSI has significantly better performance than VSM. Yalcinkaya et al. [2] presented LSI to classify patterns and trends of Building Information Modeling (BIM) by applied to 975 abstracts from academic paper with twelve classes of principal areas of research. The results show that keywords and phrases are limited. Because they only include phrases in the title, abstracts and keywords of BIM. Moreover, their corpus may consist of some other set of keywords. Shen et al. [3] have proposed a new model of latent semantic that use n-gram technique to find high level of contextual structure from queries and documents. The results reported that the model can significantly detect the distinctive semantic information from queries and documents. Their performance showed a significant outcome from the previous semantic models. Hofmann et al. [4], they applied LSI and SVD for factor analysis of counting data based on a statistical latent class model by using Expectation Maximization algorithm to search polysemous words. The results indicated that LSI with the algorithm significantly increase the searching performance for the polysemous words.

Some researchers have done their works on question answering systems without using LSI. Jin et al. [5] has proposed the improvement in Latent Dirichlet Allocation (LDA) to analyze Chinese medical record text, which is based on BM25 (Best Matching) mixture weights method. The objectives of this method are to introduce a method of extracting the record text as well as labelling them. Their results showed the good performance of grouping text paragraph meaning. Shan et al. [6] proposed a diabetes care system using data mining technique to recommend diabetics by linking the patient's clinical information, for example, medical records, laboratory tests and disease registries. The results told that data mining can be applied for implementing a powerful tool for clinical research. Sarrouti et al. [7] proposed a method for biomedical question types (QTs) classification by grouping types of biomedical questions in order for biomedical question answering system the experimental results reported that the method performed classification efficiently. Although their works improved the performance of choosing the answers, the problems involved synonyms still not be solved.

To alleviate the synonym issues, this paper presents a conceptual framework of a question answering system for diabetes care using LSI. The terms in documents are weighted by TF-IDF because it is good at extracting exact important terms from document.

The rest of the paper is organized as follows. Section 2 introduces related works, including TF-IDF weighting, LSI and singular value decomposition (SVD). The next section expresses details of the design conceptual framework, evaluation method. Finally, the conclusion is made in the last section.

2. Related Work

2.1 TF-IDF Weighting

To represent term importance in documents for text processing, it is necessary to weight the terms in the documents. There are several ways in weighting terms. The basic term weighting consists of BF (binary frequency), TF (term frequency), BF-IDF (binary frequency-inverse document frequency), and TF-IDF (term frequency-inverse document frequency) [8]. TF-IDF weighting is applied to assess how important a word is with respect to a document in a corpus. The importance of word grows proportionally to the number of times a word appears in the document, but offsetted by the frequency of the word in the collection. Variations of the TF-IDF weighting scheme are often used in considering a document's relevance given by a user query. TF and IDF are the two criteria to measure the weight of a word within text, The TF-IDF weighting schema assigns to term a weight in document and the term weighting schemes can be expressed as:

$$W_{jk} = tf_{jk} \times idf_j \qquad (1)$$

where w_{jk} is the weight of term j in document k, tf_{jk} is the number of term j that appears in document k, and idf_j is the inverse document frequency of term j as derived in the equation

$$idf_i = \log \frac{N}{df_j} \qquad (2)$$

where N is the total number of documents in the corpus, and df_j is the number of document in which the index term j appears [9-10].

2.2 Latent Semantic Indexing (LSI) and Singular Value Decomposition (SVD)

2.2.1 LSI is one of the techniques that applied in the finding process for latent semantic that based on statistic method by considering co-occurrence to define relation between keyword and document for indexing [11]. Other than considering the term frequency on the document, it also considers the relation between word with other document in the same collection as well.

2.2.2 SVD is an algorithm for decompose matrices into 3 matrices, which has principle to define latent semantic in order to reduce the matrix size of document [12]. The equation shown in the following (3) and (4).

$$\text{SVD of document } A = USV^t \tag{3}$$

$$\text{SVD of query} \quad q = q^tU_kS_k^{-1} \tag{4}$$

A is matrix of document, q is matrix of question

$U = AVS^{-1}$ is computed the original matrix can be recovered as $A = USV^t$

S is a diagonal matrix containing the singular values of the matrix A.

V is the eigenvectors of A^tA are obtained and its transpose, V^t, computed.

k is rank-k approximation for reduced matrix size

3. The Design of Framework

The design framework in this paper can be divided into three main steps. It consists of (1) document pre-processing, (2) LSI answer scoring, and (3) question answer matching, as shown in Figure 1. For the second step-LSI answer scoring, we use the LSI for finding the latent implications of the answers to the system as outlined in Section 2.2 The details of all steps will be explained as follows. For all details in this paper, a detailed description of the concept is presented in section 3.1- 3.3

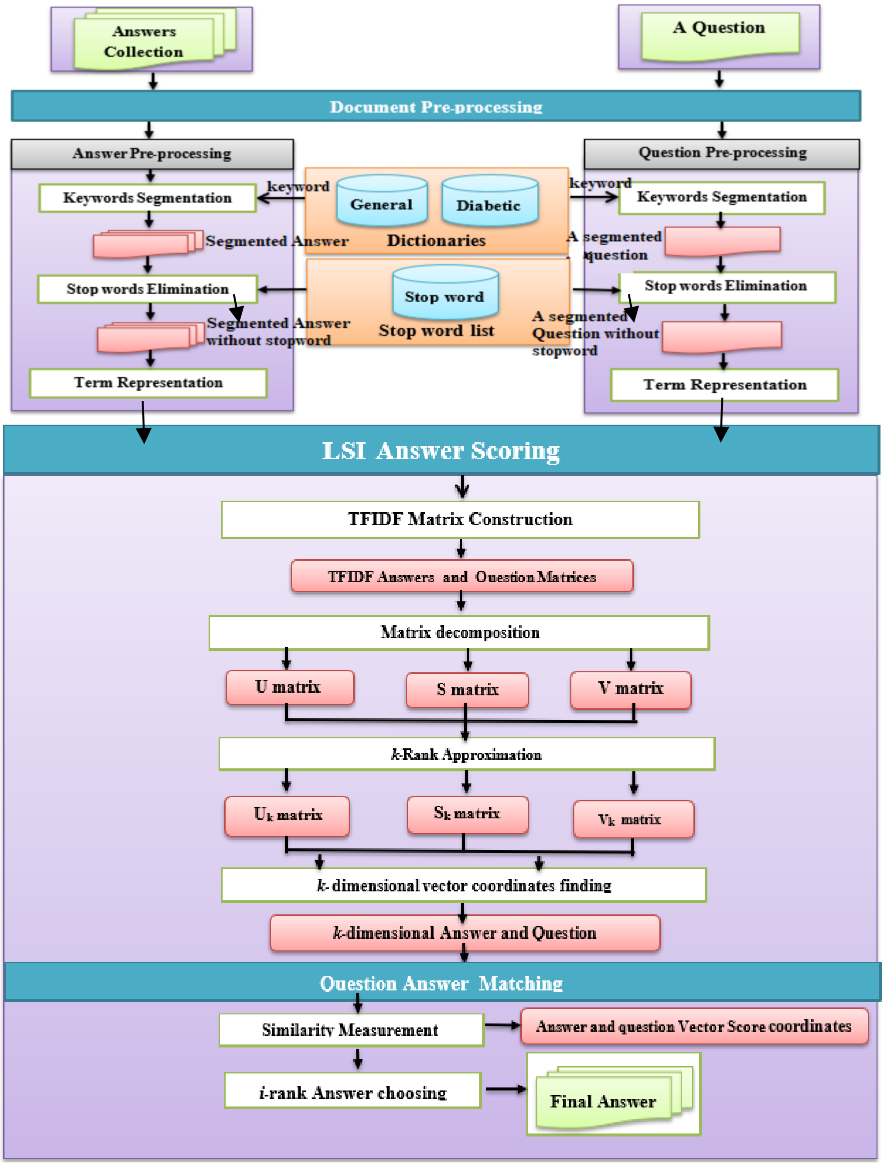
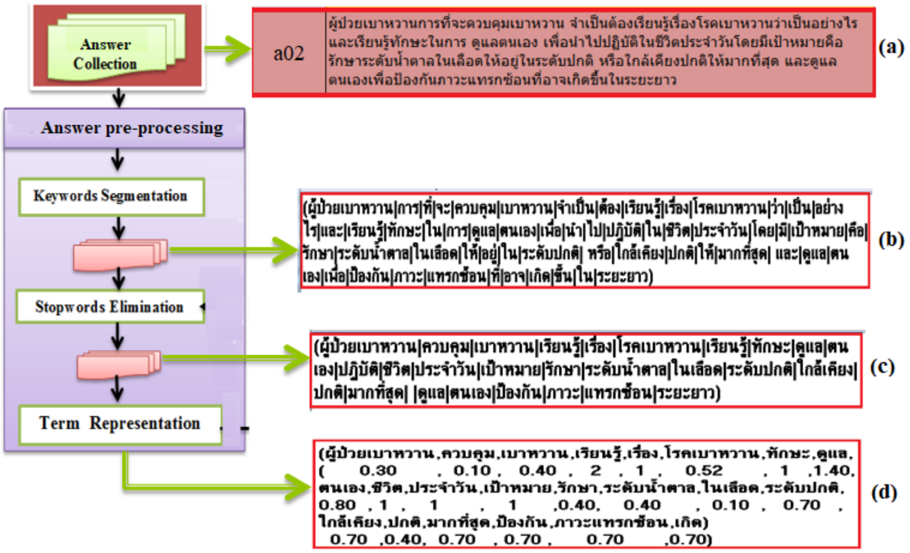


Figure 1 The Design of Framework

3.1 Document Pre-processing

This document pre-processing step is the first step in preparing the data of the answer in corpus and a question. It consists of answer pre-processing and question pre-processing. To answer pre-processing and question pre-processing, there are three common sub-processes: (1) keyword segmentation, (2) stop words elimination, and (3) term representation, as Figure 1. Figure 2 shows example of operations in the 3 sub-steps above.

For example, the word cut in answer a02 will yield the word cut off as shown in Figure 2a, 2b. The | symbol is used to separate sentences. The next step is the Stop word Elimination step. This step is a word wrapping process where unnecessary words are excluded from the sentence. For example, in Figure (2c), as a result of the termination of a sentence in sentence (2b), we will see that the word cut off, such as "ดังนั้น", "จึง", we will get a sentence without stopping words finally. The last step is to do the term representation. In this paper we chose TF-IDF weighting with LSI and SVD. The previous work we proposed four weight comparison methods BF, TF, BF-IDF and TF-IDF for question answering system in diabetes care [9]. The results showed that TF-IDF was the most effective when compared to other weight and many researchers used TF-IDF to analyze the weight of words [13-14]. After completion, we will get TF-IDF of each term as shown in Figure (2d).



ผู้ป่วยเบาหวานการที่จะควบคุมเบาหวาน จำเป็นต้องเรียนรู้เรื่องโรคเบาหวานว่าเป็นอย่างไร และเรียนรู้ทักษะในการ ดูแลตนเอง เพื่อนำไปปฏิบัติในชีวิตประจำวันโดยมีเป้าหมายคือ รักษาระดับน้ำตาลในเลือดให้อยู่ในระดับปกติ หรือใกล้เคียงปกติให้มากที่สุด และดูแลตนเองเพื่อป้องกันภาวะแทรกซ้อนที่อาจเกิดขึ้นในระยะยาว

(ผู้ป่วยเบาหวาน|การ|ที่|จะ|ควบคุม|เบาหวาน|จำเป็น|ต้อง|เรียนรู้|เรื่อง|โรคเบาหวาน|ว่า|เป็น|อย่างไร|และ|เรียนรู้|ทักษะ|ใน|การ|ดูแล|ตนเอง|เพื่อ|นำ|ไป|ปฏิบัติ|ใน|ชีวิต|ประจำวัน|โดย|มี|เป้าหมาย|คือ|รักษา|ระดับน้ำตาล|ในเลือด|ให้|อยู่|ใน|ระดับปกติ| หรือ|ใกล้เคียง|ปกติ|ให้|มากที่สุด| และ|ดูแล|ตนเอง|เพื่อ|ป้องกัน|ภาวะ|แทรกซ้อน|ที่|อาจ|เกิด|ขึ้น|ใน|ระยะยาว)

(ผู้ป่วยเบาหวาน|ควบคุม|เบาหวาน|เรียนรู้|เรื่อง|โรคเบาหวาน|เรียนรู้|ทักษะ|ดูแล|ตนเอง|ปฏิบัติ|ชีวิต|ประจำวัน|เป้าหมาย|รักษา|ระดับน้ำตาล|ในเลือด|ระดับปกติ|ใกล้เคียง|ปกติ|มากที่สุด| |ดูแล|ตนเอง|ป้องกัน|ภาวะ|แทรกซ้อน|ระยะยาว)

(ผู้ป่วยเบาหวาน,ควบคุม,เบาหวาน,เรียนรู้,เรื่อง,โรคเบาหวาน,ทักษะ,ดูแล,
(0.30 , 0.10 , 0.40 , 2 , 1 , 0.52 , 1 ,1.40,
ตนเอง,ชีวิต,ประจำวัน,เป้าหมาย,รักษา,ระดับน้ำตาล,ในเลือด,ระดับปกติ,
0.80 , 1 , 1 , 1 ,0.40, 0.40 , 0.10 , 0.70 ,
ใกล้เคียง,ปกติ,มากที่สุด,ป้องกัน,ภาวะแทรกซ้อน,เกิด)
0.70 ,0.40, 0.70 , 0.70 , 0.70 ,0.70)

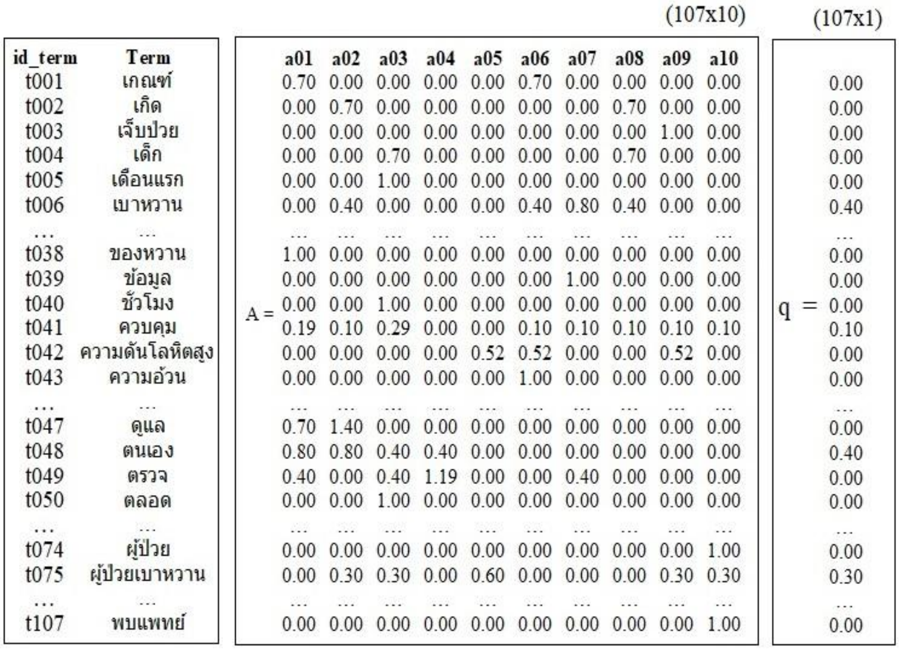
Figure 2 An example of document pre-processing step

3.2 LSI Answer Scoring

This procedure creates a term-document matrix of all answers and a matrix of questions. By LSI method, we calculate the new term scores using TF-IDF with SVD. Details and examples are as follows.

3.2.1 TF-IDF Matrix Construction

For this step It is to create a matrix of all the answers in the answer arrays (A) and the matrix of questions (q), where q is the matrix of one question. The size is mx1 (1 represents the number of questions), (m is the total number of terms in the corpus), A is the matrix of all the answers in the collection (a01-a10). The matrix A value is derived from the TF-IDF of the term in the corpus as discussed in Section 3.1. For example, Figure 3 shows the term-document of A with 10 answers and a matrix of q questions. The matrix A is 107x10, since the matrix has a total term of 107 words resulting from the word wrapping process, as described in section 3.1 (t001-t107).



(107x10) (107x1)

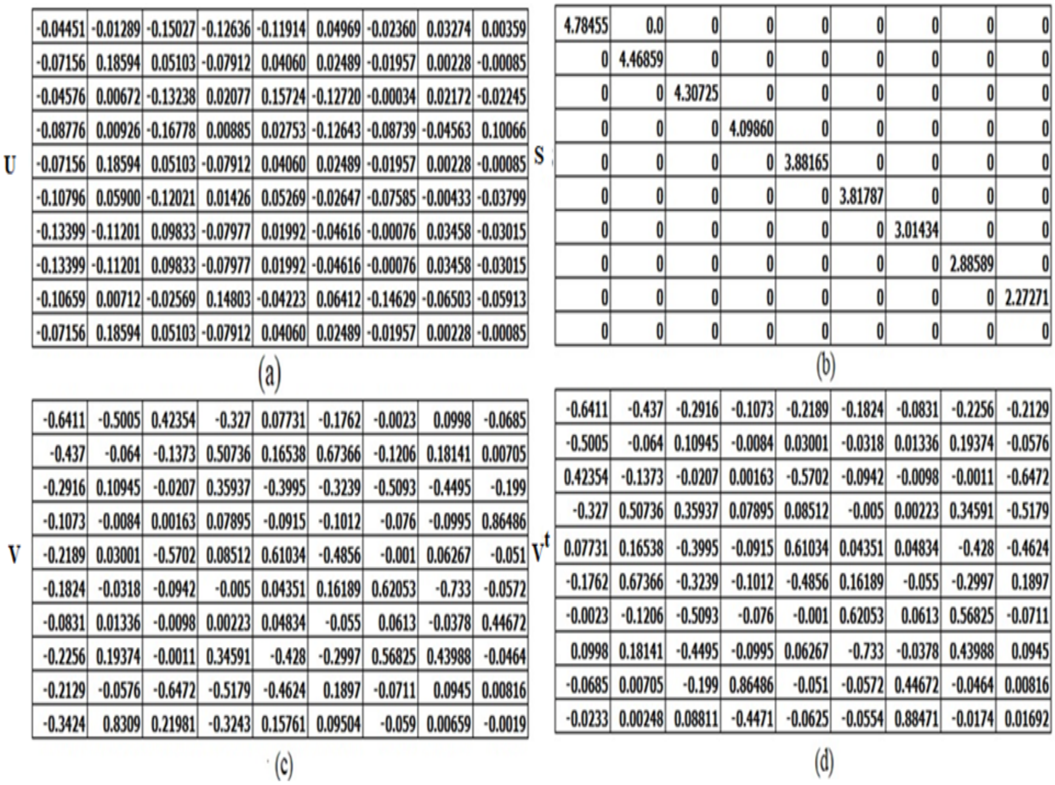
A = q =

id_term	Term	a01	a02	a03	a04	a05	a06	a07	a08	a09	a10	
t001	เกณฑ์	0.70	0.00	0.00	0.00	0.00	0.70	0.00	0.00	0.00	0.00	0.00
t002	เกิด	0.00	0.70	0.00	0.00	0.00	0.00	0.00	0.70	0.00	0.00	0.00
t003	เจ็บป่วย	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
t004	เด็ก	0.00	0.00	0.70	0.00	0.00	0.00	0.00	0.70	0.00	0.00	0.00
t005	เดือนแรก	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
t006	เบาหวาน	0.00	0.40	0.00	0.00	0.00	0.40	0.80	0.40	0.00	0.00	0.40
...	...	...	...	...	...	...	...	...	...	...	...	...
t038	ของหวาน	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
t039	ข้อมูล	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
t040	ชั่วโมง	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
t041	ควบคุม	0.19	0.10	0.29	0.00	0.00	0.10	0.10	0.10	0.10	0.10	0.10
t042	ความดันโลหิตสูง	0.00	0.00	0.00	0.00	0.52	0.52	0.00	0.00	0.52	0.00	0.00
t043	ความอ้วน	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...
t047	ดูแล	0.70	1.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
t048	ตนเอง	0.80	0.80	0.40	0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.40
t049	ตรวจ	0.40	0.00	0.40	1.19	0.00	0.00	0.40	0.00	0.00	0.00	0.00
t050	ตลอด	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...
t074	ผู้ป่วย	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
t075	ผู้ป่วยเบาหวาน	0.00	0.30	0.30	0.00	0.60	0.00	0.00	0.00	0.30	0.30	0.30
...	...	...	...	...	...	...	...	...	...	...	...	...
t107	พบแพทย์	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00

Figure 3 An example of TF-IDF matrix construction

3.2.2 Matrix Decomposition

Matrix A is decomposed three matrices (i.e., matrix U, matrix S, and matrix V) in this process. We calculate U, S and V using equations (3), (4) as described in Section 2.2. For example, we calculate U, where U = AVS-1, U is an orthogonal matrix, which U represents the keyword. Shown as Figure 4a. The keywords are sorted according to the importance of the answer along the diagonal of the S matrix. The S matrix indicates the importance of the keyword (10x10) as shown in Figure 4b. V is the eigenvectors of A^tA, it is a orthogonal as shown in Figure 4c. V^t is the transpose of the matrix with the column of V, which is the size (10x10) in Figure 4d.



U

-0.04451	-0.01289	-0.15027	-0.12636	-0.11914	0.04969	-0.02360	0.03274	0.00359
-0.07156	0.18594	0.05103	-0.07912	0.04060	0.02489	-0.01957	0.00228	-0.00085
-0.04576	0.00672	-0.13238	0.02077	0.15724	-0.12720	-0.00034	0.02172	-0.02245
-0.08776	0.00926	-0.16778	0.00885	0.02753	-0.12643	-0.08739	-0.04563	0.10066
-0.07156	0.18594	0.05103	-0.07912	0.04060	0.02489	-0.01957	0.00228	-0.00085
-0.10796	0.05900	-0.12021	0.01426	0.05269	-0.02647	-0.07585	-0.00433	-0.03799
-0.13399	-0.11201	0.09833	-0.07977	0.01992	-0.04616	-0.00076	0.03458	-0.03015
-0.13399	-0.11201	0.09833	-0.07977	0.01992	-0.04616	-0.00076	0.03458	-0.03015
-0.10659	0.00712	-0.02569	0.14803	-0.04223	0.06412	-0.14629	-0.06503	-0.05913
-0.07156	0.18594	0.05103	-0.07912	0.04060	0.02489	-0.01957	0.00228	-0.00085

(a)

S

4.78455	0.0	0	0	0	0	0	0	0
0	4.46859	0	0	0	0	0	0	0
0	0	4.30725	0	0	0	0	0	0
0	0	0	4.09860	0	0	0	0	0
0	0	0	0	3.88165	0	0	0	0
0	0	0	0	0	3.81787	0	0	0
0	0	0	0	0	0	3.01434	0	0
0	0	0	0	0	0	0	2.88589	0
0	0	0	0	0	0	0	0	2.27271
0	0	0	0	0	0	0	0	0

(b)

V

-0.6411	-0.5005	0.42354	-0.327	0.07731	-0.1762	-0.0023	0.0998	-0.0685
-0.437	-0.064	-0.1373	0.50736	0.16538	0.67366	-0.1206	0.18141	0.00705
-0.2916	0.10945	-0.0207	0.35937	-0.3995	-0.3239	-0.5093	-0.4495	-0.199
-0.1073	-0.0084	0.00163	0.07895	-0.0915	-0.1012	-0.076	-0.0995	0.86486
-0.2189	0.03001	-0.5702	0.08512	0.61034	-0.4856	-0.001	0.06267	-0.051
-0.1824	-0.0318	-0.0942	-0.005	0.04351	0.16189	0.62053	-0.733	-0.0572
-0.0831	0.01336	-0.0098	0.00223	0.04834	-0.055	0.0613	-0.0378	0.44672
-0.2256	0.19374	-0.0011	0.34591	-0.428	-0.2997	0.56825	0.43988	-0.0464
-0.2129	-0.0576	-0.6472	-0.5179	-0.4624	0.1897	-0.0711	0.0945	0.00816
-0.3424	0.8309	0.21981	-0.3243	0.15761	0.09504	-0.059	0.00659	-0.0019

(c)

V^t

-0.6411	-0.437	-0.2916	-0.1073	-0.2189	-0.1824	-0.0831	-0.2256	-0.2129
-0.5005	-0.064	0.10945	-0.0084	0.03001	-0.0318	0.01336	0.19374	-0.0576
0.42354	-0.1373	-0.0207	0.00163	-0.5702	-0.0942	-0.0098	-0.0011	-0.6472
-0.327	0.50736	0.35937	0.07895	0.08512	-0.005	0.00223	0.34591	-0.5179
0.07731	0.16538	-0.3995	-0.0915	0.61034	0.04351	0.04834	-0.428	-0.4624
-0.1762	0.67366	-0.3239	-0.1012	-0.4856	0.16189	-0.055	-0.2997	0.1897
-0.0023	-0.1206	-0.5093	-0.076	-0.001	0.62053	0.0613	0.56825	-0.0711
0.0998	0.18141	-0.4495	-0.0995	0.06267	-0.733	-0.0378	0.43988	0.0945
-0.0685	0.00705	-0.199	0.86486	-0.051	-0.0572	0.44672	-0.0464	0.00816
-0.0233	0.00248	0.08811	-0.4471	-0.0625	-0.0554	0.88471	-0.0174	0.01692

(d)

Figure 4 An example of U, S, and V matrices

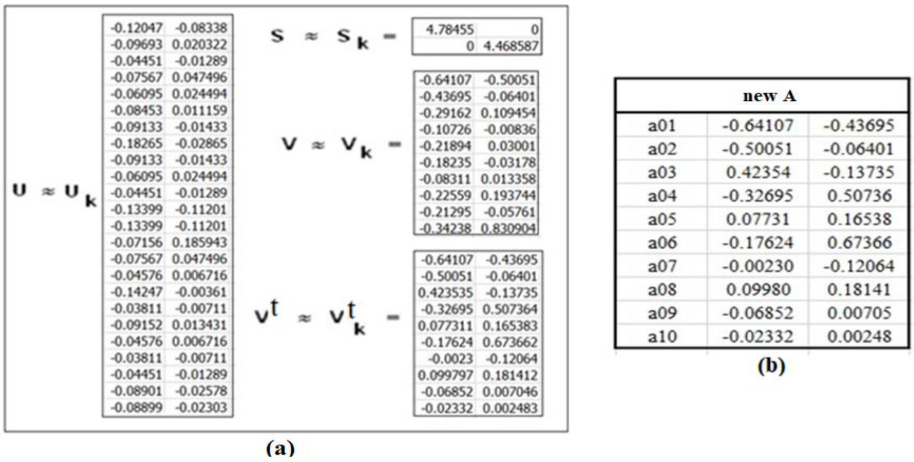
3.2.3 *k*- Rank Approximation

This step is an estimate rank of k rank for the matrix U, S, V, which k is the Rank-k approximation for reducing matrix size.

For example, the U, S, and V matrices are shown in step 3.2.2, if k = 2, U_k is the matrix that contains the first 2 columns of U matrix, S_k is the result that consists of first 2 columns of S, V_k is the result that consists of first 2 columns of V, V^t is the transpose of a matrix, those rows are the columns of V matrix, V^t_k is the result that contains the first 2 columns of V^t as shown in Figure 5a.

3.2.4 *k*-Dimensional Vector Coordinates Finding

3.2.4.1 Procedure is the finding process for the new matrix of the answer (new A) in the corpus. We use equation A = USV^t as shown in section 2.2 in order to reduce the size of the *k*-dimension by considering from v^t_k matrix. As show in step 3.2.3. For example, if k = 2, V_k is the result that consists of the first 2 columns in the matrix, which the matrix size (10x2) is shown in Figure 5b.



$U \approx U_k$

-0.12047	-0.08338
-0.09693	0.020322
-0.04451	-0.01289
-0.07567	0.047496
-0.06095	0.024494
-0.08453	0.011159
-0.09133	-0.01433
-0.18265	-0.02865
-0.09133	-0.01433
-0.06095	0.024494
-0.04451	-0.01289
-0.13399	-0.11201
-0.13399	-0.11201
-0.07156	0.185943
-0.07567	0.047496
-0.04576	0.006716
-0.14247	-0.00361
-0.03811	-0.00711
-0.09152	0.013431
-0.04576	0.006716
-0.03811	-0.00711
-0.04451	-0.01289
-0.08901	-0.02578
-0.08899	-0.02303

$S \approx S_k =$

4.78455	0
0	4.468587

$V \approx V_k =$

-0.64107	-0.50051
-0.43695	-0.06401
-0.29162	0.109454
-0.10726	-0.00836
-0.21894	0.03001
-0.18235	-0.03178
-0.08311	0.013358
-0.22559	0.193744
-0.21295	-0.05761
-0.34238	0.830904

$v^t \approx v^t_k =$

-0.64107	-0.43695
-0.50051	-0.06401
0.423535	-0.13735
-0.32695	0.507364
0.077311	0.165383
-0.17624	0.673662
-0.0023	-0.12064
0.099797	0.181412
-0.06852	0.007046
-0.02332	0.002483

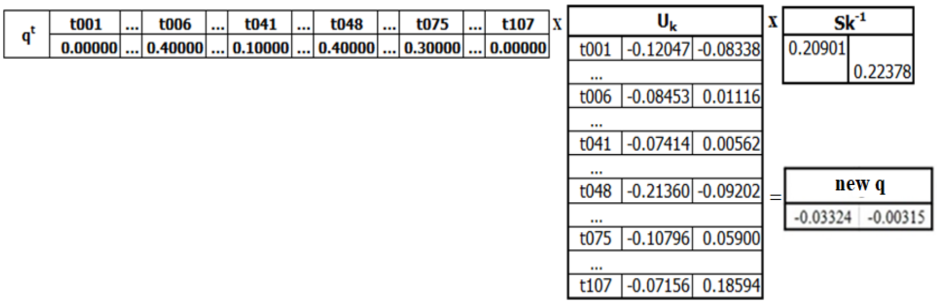
(a)

new A		
a01	-0.64107	-0.43695
a02	-0.50051	-0.06401
a03	0.42354	-0.13735
a04	-0.32695	0.50736
a05	0.07731	0.16538
a06	-0.17624	0.67366
a07	-0.00230	-0.12064
a08	0.09980	0.18141
a09	-0.06852	0.00705
a10	-0.02332	0.00248

(b)

Figure 5 An example of U_k, V_k, and S_k matrices

3.2.4.2 This step is to find the new matrix for question (new q), 1 question in order to reduce the size of *k*-dimension by using equation q = $q^tU_kS_k^{-1}$ as mentioned in section 2.2. For example, if k = 2, U_k is the matrix that contains the first 2 columns of the U matrix, S_k is the matrix that contains the first 2 columns of the S matrix. The matrix size (10x2) is shown in Figure 6.



q^t	t001	...	t006	...	t041	...	t048	...	t075	...	t107
	0.00000	...	0.40000	...	0.10000	...	0.40000	...	0.30000	...	0.00000

X

U_k		
t001	-0.12047	-0.08338
...		
t006	-0.08453	0.01116
...		
t041	-0.07414	0.00562
...		
t048	-0.21360	-0.09202
...		
t075	-0.10796	0.05900
...		
t107	-0.07156	0.18594

X

Sk^{-1}	
0.20901	
	0.22378

=

new q	
-0.03324	-0.00315

Figure 6 An example of k-dimensional question vector coordinates

3.3 Question Answer Matching

This process is the matching of the weight of keywords in the question and answer in the document. It considers the similarity between the weight of keywords in the question and the keyword of answer in the document. There are two steps as following.

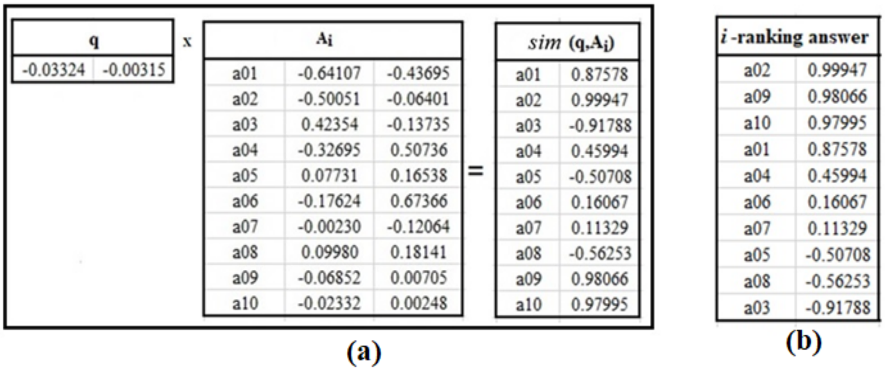
3.3.1 Similarity Measurement

This step is to measure the similarities between question and answer by using cosine similarities [15], which define q and A are vector that is able to calculate as shown in Figure 7a. Assigned q and A is vector; q is question, A is document of answer, *i* is sequence of answer then the result will be calculated by the following equation.

$$sim(q, A) = \frac{q,A}{[q][A]} \qquad (5)$$

3.3.2 *i*-Rank Answer Choosing

It is the sorting of the answer step, which is determined from the result in section 3.3.1 by sorting results in descending order as showed in Figure 7b.



q	
-0.03324	-0.00315

x

Ai		
a01	-0.64107	-0.43695
a02	-0.50051	-0.06401
a03	0.42354	-0.13735
a04	-0.32695	0.50736
a05	0.07731	0.16538
a06	-0.17624	0.67366
a07	-0.00230	-0.12064
a08	0.09980	0.18141
a09	-0.06852	0.00705
a10	-0.02332	0.00248

=

sim (q,A*i*)	
a01	0.87578
a02	0.99947
a03	-0.91788
a04	0.45994
a05	-0.50708
a06	0.16067
a07	0.11329
a08	-0.56253
a09	0.98066
a10	0.97995

(a)

i -ranking answer	
a02	0.99947
a09	0.98066
a10	0.97995
a01	0.87578
a04	0.45994
a06	0.16067
a07	0.11329
a05	-0.50708
a08	-0.56253
a03	-0.91788

(b)

Figure 7 An example of calculating a similarity between q and A$_i$

4. Evaluation Method

To evaluate our proposed framework, we have three experts to check answers which are suggested from the system. If two of the experts agree on the result, they will be considered as a correct suggestion. If there is only one or no one coincides with the result, the suggested answer will be accounted as a wrong suggestion. The performance of the whole system will be assessed by an accuracy which refers to the closeness of a measured result by the system to expert answers.

5. Conclusions and Future Works

This paper proposed a framework of question answering systems for diabetes care using LSI with text mining pre-processing. We explained steps in the framework and exemplified the steps by showing a user query and its answers using LSI. The result of the preliminary experiment showed that the conceptual framework provided the correct answer. The concept of this framework can be applied to other domains in question answer systems. The limitation of the conceptual framework is the processes of keyword segmentation and stop word elimination, as if the keyword is not in the dictionary, the system cannot match the keyword between the used query and the candidate answers in a collection. For future works, we plan to ensure the proposed framework by making a full experiment with a dataset of more real questions and answers.

Acknowledgement

This work was also supported by Rajamangala University of Technology Srivijaya.

References

[1] Al-Anzi FS, AbuZeina D. Toward an enhanced Arabic text classification using cosine similarity and Latent Semantic Indexing. Journal of King Saud University-Computer and Information Sciences 2017;29(2):189-95.

[2] Yalcinkaya M, Singh V. Patterns and trends in building information modeling (BIM) research: a latent semantic analysis. Automation in Construction 2015;59:68-80.

[3] Shen Y, He X, Gao J. Deng L, Mesnil G. A latent semantic model with convolutional-pooling structure for information retrieval. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management; 2014 November 3-7. p. 101-10.

[4] Hofmann T. Probabilistic latent semantic indexing. In ACM SIGIR Forum 2017;51(2): 211-8.

[5] Jin X, Jin Q, Li Y. A Method of automatic annotation for medical record text based on latent dirichlet allocation. Proceedings of International Conference on Electromechanical Control Technology and Transportation (ICECTT 2015); 2015. p. 305-8

[6] Shah BR, Lipscombe L. Clinical diabetes research using data mining: a Canadian perspective. Canadian journal of diabetes 2015;39(3):235-8.

[7] Sarrouti M, Lachkar A, Ouatik SEA. Biomedical question types classification using syntactic and rule based approach. In 2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K); 2015 Nov 12-14; Lisbon, Portugal. IEEE; 2015. p. 265-72.

[8] Kittiphattanabawon N, Theeramunkong T, Nantajeewarawat E. Region-based association measures for ranking mined news relations. Intelligent Data Analysis 2014;18(2):217-41.

[9] PAISI. Proceedings of Intelligence and security informatics: Pacific Asia workshop, PAISI 2010; 2010 June 21; Hyderabad, India. n.p.: Springer.

[10] Salton G. Automatic processing of foreign language documents. Journal of the Association for Information Science and Technology 1970;21:187-194.

[11] Landauer TK. Latent semantic analysis. John Wiley & Sons; 2006.

[12] Baker K. Singular value decomposition tutorial. The Ohio State University 2005;24.

[13] Phetkrachang K, Kittiphattanabawon N. Thai question answering systems in diabetes using logical co-operators. The Twelfth 2017 International Conference on Knowledge, Information and Creativity Support Systems (KICSS2017); 2017 Nov 9-11; the Nagoya Institute of Technology, Nagoya, Japan.

[14] Lowe R, Pow N, Serban I, Pineau J. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. 2015;1506.08909.

[15] Alodadi M, Janeja VP. Similarity in patient support forums using tf-idf and cosine similarity metrics. In 2015 International Conference on Healthcare Informatics (ICHI); 2015 Oct 21-23; Dallas, TX, USA. IEEE;2015. p.521-2.

Author's Profile

Ketsara Phetkrachang received a bachelor and master degree in Mangement of Information of technology from Ramkhamhaeng University, Thailand in 1995 and Walailak University, Thailand in 2003, respectively. She is a doctoral student, School of Informatics, Management of Information Technology, Walailak University. Her research interests are Information Retrieval, Information Engineering, Question answering systems and Expert System.

Nichnan Kittiphatttanabawon received a bachelor and master degree in Computer Science from Rangsit University in 1993 and Prince of Songkla University in 1999, respectively, and doctoral degree in Technology (International Program) from Sirindhorn International Institute of Technology, Thammasat University in 2012. Currently, she is a lecturer in department of information technology, school of informatics, Walailak University. Her current research interests include text mining, data mining, document relation discovery, and association analytics.