Improving the Retrieval Performance by Using the Distance-Based Bigrams

Pakinee Aimmanee[1] and **Thanaruk Theeramunkong**[2], Non-members

ABSTRACT

In this paper, we discuss a new way to improve retrieval performance using a special type of bigrams called Distance-based Bigram (DB). DB is a word pair whose distance between the two components is greater than or equal to one. DB allows us to find documents that express a phrase or a sentence differently from a query. The results show that DB combined with unigram performs significantly better than the unigram and bigram when the first few correct documents are needed.

Keywords: Distance-Based Bigram, Vector Space Model, Information Retrieval

1. INTRODUCTION

The rapid development of computer and the Internet technology has created worldwide network connections and communications. This rapid growth of technology causes a big change in the usage of the Internet. Many websites has been created as the media to present valuable information in several fields such as education, sciences, engineering, and medicines to educate and to make convenience for users. Especially for the field of the medicines, it allows the users to search and get their needed information about symptoms and diseases before go to consult with a real doctor. Although, a medical collection is a field that requires a very high accuracy on the retrieval results, the results are usually not very accurate. The following three barriers are blamed to be the causes of the poor performance. The first barrier is synonyms. The synonyms occur frequently in the medical field because medical words derived from many languages and sources such as local names, scientific names, and commercial product names. As a result, for each medicinal jargon term, it may appear in many different forms that have the same meaning. Such words are called synonyms. The second barrier is the large overlapping of words among diseases. As a result, when we search for one thing, the search engine may falsely retrieve another one that is totally unrelated. The last barrier is language problems. In some languages like Chinese and Thai, there are many compound words and there is no word boundary. A compound word comes from multiple words that are joined together forming a new word whose meaning may or may not be related with their original words. In some system that handles a search keyword by segmenting them first before searching, it may retrieve the documents related to its two sub-words and get the several unrelated answer. In general, there are two main approaches for searching documents; string matching and word matching [1][2]. For the string matching, one tries to find a set of documents that exactly matches with the input query. The retrieval results are usually accurate with high precision but low recall. That is the returned results obtained from this approach are usually relevant with the query (high precision) but it may not be possible to find all relevant documents (low recall) as the pattern is too specific. For instance, when a modifier is added to or removed from a query, a relevant document that is added with modifiers is not been selected. As another method, word matching does not strictly use the exact order of the input keyword. The input keyword is separated into a set of shorter units before querying. Since Thai language has no word boundary indication, it requires an extra step to segment a sentence into a series of words, called word segmentation process. Thus, the accuracy of the word matching depends on the parsing scheme used. As a consequence, the results from a language that has no word boundary may be not as accurate as that of English. It has low accuracy because a word may be a substring of another. For example, a Thai word เบาหวาน (diabetic) contains two substrings เบา(not heavy) and หวาน(sweet) which can be recognized as a word. Thus, if someone searches for this keyword, not only the documents about diabetics that this person would obtain, he or she may also obtain documents about เบา(not heavy) or หวาน(sweet) instead. Besides the sub-word problem, the word matching also faces with the problem of word order, or alternative expression as it does not take the position into account which still is a drawback from this method. A typical problem for word matching can be illustrated by two sentences กิน(take) ยา (medicine) เพราะ (because) ป่วย(sick) meaning take medicine because of sickness and ป่วย(sick) เพราะ (because) กิน (take) ยา(medicine) meaning sick because of taking medicine. These two sentences would mean the same thing though seman-

Manuscript received on August 1, 2009 ; revised on October 29, 2009.

[1,2] The authors are with The School of Information, Computer, and Communication Technology
Sirindhorn International Institute of Technology
Thammasat University, Thailand , E-mail: pakinee@siit.tu.ac.th and thanaruk@siit.tu.ac.th

tically they are opposite. In this paper, we introduced a new scheme that can be used for solving the problems occurred by string matching and word matching to improve the search efficiency. The paper is organized as follows: in Section 2, we formally describe distance based bigram that are to be used to improve and to solve language problems; in Section 3, we describe our framework; in Section 4, we describe results and discussion; finally in Section 5, we conclude our work in the summary.

2. DISTANCE-BASED BIGRAMS

This section describes a concept of distance-based bigrams. In the fields of information retrieval and text categorization, a document is usually presented in the form of bag-of-words, i.e., a set of words that are usually come with their frequencies [2]. The most commonly used types of bag-of-words are n-gram. When n=1 and n=2, they form two well known bag-of-word types so called unigrams and bigrams, respectively. Unigrams are single words, usually smallest units of a sentence, while bigrams compose of two consecutive unigrams. In English, it is easy to separate each individual word from a string of text since there are explicit boundary markers (usually space) among them. However, for Asian languages such as Thai and Chinese, a text composes of words concatenating with each other without any boundary marker. Therefore, building unigrams from a Thai running text need an extra process called word segmentation. The unigrams are often used in most retrieval scheme because of its simplicity. However, there is a major drawback from unigrams. It contains just a bag of unigrams for which we find no relationship or order between them. One way to solve the order-matter problem is to use bigrams. Each bigram keeps the order of two sub-unigrams according to their appearance in the original text input. For example, when bigrams are applied to these two sentences "vaccines defeat germs", and "germs defeat vaccines," the first one produces bigrams "vaccines defeat" and "defeat germs" whereas the second one produces "germs defeat" and "defeat vaccines." Therefore, these two sentences have different sets of bigrams. The bigrams can sometimes be used to represent compound words for languages that use two words to form a new word. When keywords are longer than two units, n-grams with n greater than two such as trigrams and tetragrams shall be applied. Nonetheless, bigrams can only cover pairs of two words that are immediately appears next to each other. In this work, we propose a new way to look at bigrams so-called distance-based bigram to build a bag-of-word for bigrams. This distance-based bigram expands varieties of bigrams according to the distance from the first unigram in the scope of n-unigrams. Its formal definition is as below.

Definition:(DB_n) *Given an n-consecutive unigrams* $u_i u_{i+1} u_{i+2} ... u_{i+n}$ *in the text, N-distance-based bigrams* (DB_n) *is a set of distance-based bigrams whose distance between each sub-unigram is less than n. Mathematically, it can be described as*

$$DB_n = \{u_a u_b | 0 < b - a \leq n\}. \quad (1)$$

DB allows a search engine to finds more pairs of unigrams (terms) that appears not only consecutively but also in the same neighborhood. Here is an example. If we have a Thai text โรค ที่ เกิด จาก พยาธิ (diseases that are caused by parasites) the Db_4 = { โรค-ที่, โรค-เกิด, โรค-จาก, โรค-พยาธิ, ที่-เกิด, ที่-จาก, ที่-พยาธิ, เกิด-จาก, เกิด-พยาธิ, จาก-พยาธิ} . Compared to the standard bigrams, there are only โรค-ที่, ที่-เกิด, เกิด-จาก, จาก-พยาธิ. The distance-based bigram have more varieties than the conventional bigrams. DB is derived from the concept of s-gram which is proposed by PirKola and Keskustalo (2002) for generating cross- and mono-lingual form variants [3]. The s-gram is defined as a sequence of n grams that allows some grams inside of the sequence to be omitted. It has been also recognized as gapped q-gram which is proposed by Burkhardt and Karkainen (2003) for quick and efficient filtering for approximate string matching [4]. For the gapped q-gram, the pattern of interested character string is described as a shape $\#*\#*\#\#$ where the $*$ can be any character and the sharp sign (#) means a fixed character. Both q-grams and s-grams were defined and used in the character level. In our scheme, we used a variety of gapped bigrams in the word level that are of shapes $\#\#$, $\#*\#$, $\#**\#$, $\#***\#$ to create a variety of terms for improving the search retrieval.

3. THE PROPOSED FRAMEWORK

In this section, our framework on indexing terms using DB is described. Composed of four steps, the indexing and querying processes are performed. The details are given below.

- As the initial stage, the documents and queries are prepared and set up.
- In the indexing process, indices for the documents are created. In this work, five major schemes are used: unigram, bigram, bigram with unigram, DB, and DB with unigram. Each term type is used in forming of a vector.
- In the querying process, each query is converted to a vector using the above-mentioned schemes. The term weightings are defined for these processes.
- To retrieve documents that are the most related to the query, we compute the similarity among them. The similarity measure used in this work is cosine similarity. Each step can be explained in more detail in the next subsection.

3.1 Document and Query Preparation

We used two document collections that are alike in size, context, and characteristics. The first document collection contains 1,000 documents that are sampled from a Thai Medical corpus, composed of 10,567 documents taken from several major Thai medical websites. For simplicity, we call this collection MD1000. This collection of documents comprises general information about diseases, causes, symptoms, cautions, preventions, and treatments which. Twelve queries are created, and their related documents are collected manually by reading these documents. The second document collection is MEDLINE which is obtained from abstracts of health and medical journals. It contains 1,033 documents relating to diseases, anatomy, and pharmaceuticals. This collection is all written in English. The specifications and details of document collections are summarized in Table 1.

Table 1: *Specifications of document collections used in our experiment*

Specifications	MD1000	MEDLINE
Language	Thai	English
Number of documents	1,000	1,033
Number of distinct words	6,356	9,168
Average number of words per document	99.66	79.58
Number of queries	12	30
Average number of relevant documents per query	6.25	22.36
Average number of words per query	22	11

General text preprocesses such as case folding and removing stop words for documents and the queries of the collection are applied to the text documents. Since Thai language has no capital letters and has no obvious suffixes to be stemmed as in English, the stemming process is not done.

/subsectionVector Formation In this section, we describe how the text documents and queries are converted into vectors. Variants of gapped bigrams that form DB are generated and studied and five types of terms: unigram, bigram, unigram with bigram, DB, bigram with unigram, and DB with unigram are used for representing a term associated with its corresponding row in the vector. Using the term frequencies weighting for the unigram and bigram, we converted the sets of document collection and theirs queries into vectors using the term frequency. The frequency weighting of each term is the number of times that it appears in the document. For DB, a weighting function defining the weight between two unigrams is provided below.

Definition:(*distancefunction*) *Given* $S = u_0u_1...u_{n-1}$ *composed of n unigrams and a distance function*

$$d(S,i,j,x,y) = \begin{cases} j-i-1 & \text{if } u_i = x \text{ and } u_j = y \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

the weight between unigrams x *and* y *in* S *containing* n *character is*

$$w(x,y) = \Sigma_{i=0}^{n-1}\Sigma_{j=i}^{n-1}\frac{1}{d(S,i,j,x,y)} \quad (3)$$

For example, if a document contains a string xyzxy, the term vectors using gapped bigram, unigram, bigram, DB, bigram with unigram, and DB with unigram are shown as below.

$$\begin{matrix} xz \\ yx \\ xy \end{matrix}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad \begin{matrix} xx \\ yy \end{matrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \begin{matrix} xy \end{matrix}\begin{pmatrix} 1 \end{pmatrix}$$

Fig.1: *Shown from left to right, the vectors generating from a string xyzxy using gapped bigram with 1, 2, and 3 skips respectively.*

$$\begin{matrix} x \\ y \\ z \end{matrix}\begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}, \quad \begin{matrix} xy \\ yz \\ zx \end{matrix}\begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}, \quad \begin{matrix} x \\ y \\ z \\ xy \\ yz \\ zx \end{matrix}\begin{pmatrix} 2 \\ 1 \\ 1 \\ 2 \\ 1 \\ 1 \end{pmatrix},$$

$$\begin{matrix} xy \\ yz \\ zx \\ xz \\ yx \\ zy \\ xx \\ yy \end{matrix}\begin{pmatrix} 2 \\ 1 \\ 1 \\ 0.5 \\ 1 \\ 0.5 \\ 0.33 \\ 0.33 \end{pmatrix}, \quad \begin{matrix} x \\ xy \\ xz \\ xx \\ y \\ yz \\ yx \\ yy \\ z \\ zx \\ zy \end{matrix}\begin{pmatrix} 2 \\ 2 \\ 0.5 \\ 0.33 \\ 2 \\ 1 \\ 1 \\ 0.33 \\ 1 \\ 1 \\ 0.5 \end{pmatrix}$$

Fig.2: *Shown from left to right, the vectors generating from a string xyzxy using unigram, bigram, unigram with bigram, DB, bigram with unigram, and DB with unigram, respectively*

For the gapped bigrams shown in Fig. 1, the term frequency of each term is one as each term appears only a single time. For Fig. 2, the term frequencies for the unigrams are quite obvious as x and y appear in the phrase twice where as z appears only one time. For a regular bigram (xy) appear twice whereas zx appear one time. For the DB with the unigram, the weights of each unigram are the frequencies of each unigram. The distance weight of xy equals to 2 as they appear next to each other twice; xz equals to 0.5 because the distance from x to z is 2; xx equals to 0.33 because the distance from first x to the second x is 3 and so on.

3.2 Vector Proximity and Evaluation

We used a well known algorithm for information retrieval called the *vector space model* [5][6] to determine which documents are relevant to the queries. The vector space model finds the similarity between a document vector and a query vector by finding the cosine similarity between them. For all gapped bigram and five types of terms, we computed the cosine similarities between all documents and queries and ranked from largest and smallest. To evaluate the results of each query, we compared them with their relevant judgment sets provided along with the document collections and measured the performance in terms of recall and precision. Their definitions are as follows.

Definition:(Recall) *For a query q, the recall denoted as R of the results is the ratio of the number of relevant documents of q that are retrieved over the number of total of relevant documents of q.*

Definition:(Precision) *For a query q, the precision denoted as P of the results is the ratio of the number of relevant documents of q that are retrieved over the number of total of documents that are currently retrieved.*

The recall measures the completeness of the solution whereas the precision indicates the accuracy. When there are multiple queries, the average recall $\bar{R}$ and average precision $\bar{P}$ are used to measure the overall performance. $\bar{R}$ is the average value of all R's from all queries and likewise $\bar{P}$ is the average value of all P's from all queries.

In our experiment, the average recalls and average precisions of the results are collected. The results of unigrams and those of the bigrams are used as the baselines for the performance comparison.

4. RESULTS AND DISCUSSIONS

This section shows results of the average recall and the average precisions when different term types are used. The result presentations are divided into two main groups. The first is the results from gapped bigrams which are used to form DB. The second is the results from five major term types.

4.1 Results from Gapped Bigrams

Shown in Table 2, the results of the gapped bigrams characterized by the number of skips (0, 1, 2, 3, and 4) used in either the documents or the queries. Fig. 3 and 4 illustrate results of the average precision at only the first top 20 ranks of MEDLINE and MD1000. Appeared in Table 2 and Fig. 3 and 4 as a pair in parenthesis, the first and second components are the number of skips of gapped bigram used in the documents and in the queries, respectively. Note that when the number of skip of both document side and query side is zero, it is a conventional bigram.

Table 2: *The average precisions and recalls in percent when all ranks($\bar{R}$ and $\bar{P}$) are considered of gapped bigrams characterized by the number of skips used in documents (sd)and queries(sq) for finding the cosine similarities.*

(s_d, s_q)	$(\bar{R},\bar{P})$ of MEDLINE	$(\bar{R},\bar{P})$ of MD1000
(0,0)	(38,28)	(88, 42)
(1,0)	(8, 60	(19, 2)
(2,0)	(5, 4)	(23,3)
(3,0)	(3,3)	(9,4)
(0,1)	(4,3)	(17,2)
(0,2)	(3,2)	(22, 4)
(0,3)	(4,3)	(24, 8)

The gapped-bigram vectors of documents are tested against a conventional bigram vector of the queries to imitate the scenario that the text documents are added with modifiers. The observations from the graphs, and their explanations and discussions are given as follow.

- According to the results in Table 2, the numerical average recalls and average precisions of MD1000 are noticeably higher than those of MEDLINE. This can be explained that Thai words are constructed by combining multiple single words all together to form a new meaning. Since medical technical terms or jargons such as the symptoms, disease's names, foreign medicine names are usually been derived or given names based on descriptions or phrases that explain its functionality, thus the compound words can be expressed differently. For example, the word "gastrointestinal diseases" are called differently in Thai as โรค (disease) ระบบ (system) ทางเดิน (path way) อาหาร(food), โรค(disease) ที่พบใน(found in) ระบบ(system) ทางเดิน(path way) อาหาร(food), and โรค(disease) ทางเดิน)path way) อาหาร(food). Adjectives or modifiers can be inserted (or removed) between (or from) the component words as a result the recalls and the precisions when gapped bigram is used are higher for MD1000 than MEDLINE.
- The recalls and the precisions of the results obtained from gapped bigrams are not as high as those of the conventional bigram ((0, 0)). This can be explained that when the documents represented by a gapped bigram are cosine similarity tested against a query represented by a conventional bigram (and vice versa), there are not many terms in common between them. In addition, we did a deep analysis on the ratio of the number of gapped bigram terms with 1, 2, and 3 skips that are intersection with the conventional bigram terms to the number of conventional bigram terms and found that the ratios are 0.49, 0.36, and 0.25, respectively for MD1000 and 0.13, 0.10, and 0.09 respectively for MEDLINE. This shows that as the numbers of skips are increasing, the number

of words that are common between gapped bigrams and conventional bigrams are decreasing both collections. This also implied that there are more varieties of gapped bigrams in MD1000 than the MEDLINE. This evident also explains the reason that the precisions of gapped bigrams of MD1000 are higher than those of the MEDLINE.

• Although the results in Table 2 from the overall performance of the gapped bigrams are not as good as the conventional bigram, there is an important observation that supports our idea that gapped bigrams help us improve the performance of IR. When we considered the results of the average precision at the first top 20 ranks of MEDLINE and MD1000 shown in Fig. 4 and 5, we observe that some results from gapped bigrams used in documents or query pairs are obviously better than the conventional bigram in term of precision, for example (1, 0) and others at the very top ranks in MEDLINE. This implies that relevant documents are denser at the very top ranks than the later ranks. For MD1000, the conventional bigram term model ((0, 0)) yields the highest precision. This result agrees with the results in Table 1 due to the great usage of compound words in Thai language. Other pairs such as (0, 1) and (1, 0) also give high precisions. Even though the results from gapped bigrams are lower than the conventional bigram, our further investigation on the list of relevant documents retrieved at the top ranks of the gapped bigrams reveals that these relevant documents often contains new relevant documents that has a lower rank in the conventional bigram. For example, one of the queries is โรค (disease)ระบบ(system)ทางเดิน(pathway) หายใจ (breathing) which means respiratory system, when the conventional bigrams are used, one of the relevant documents that uses a different phrase โรคที่พบในระบบทางเดินหายใจ are retrieved at rank 1 in the gapped bigrams where as in the conventional bigram retrieves it at rank 7.

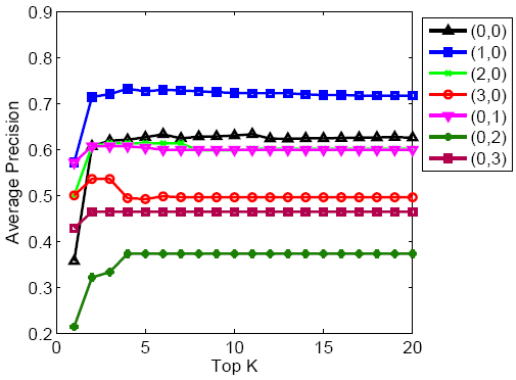

Fig.3: *The average precision of MEDLINE at the first top 20 ranks when the gapped bigrams are used in the documents or in the queries. Shown in parenthesis, first and second are the numbers of skips that are used in the documents and in the queries, respectively*

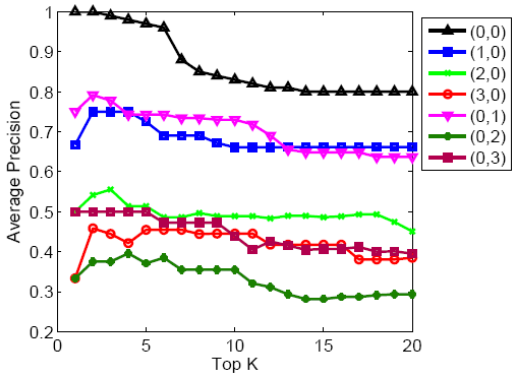

Fig.4: *The average precision of MD1000 at the first top 20 ranks when the gapped bigrams are used in the documents or in the queries. Shown in parenthesis, first and second are the numbers of skips that are used in the documents and in the queries, respectively.*

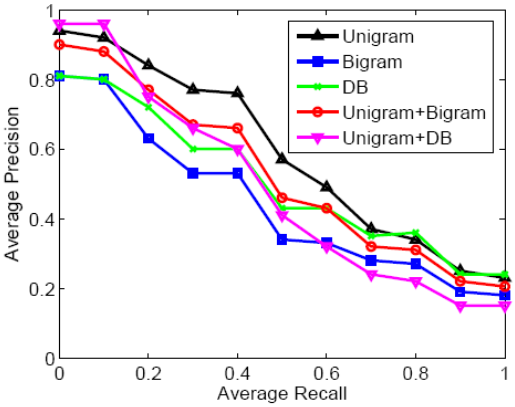

Fig.5: *The interpolated average recall ($\bar{R}$) vs. average precision ($\bar{P}$) of MD1000 for unigram, bigram, DB, bigrams with unigrams, DB with unigrams.*

4.2 Results of DB and its combinations

We used a linear combination of gapped bigrams defined in the subsection 4.1 to construct DB. In this section, we compare the results of DB and its variants of DB with other term types. Unigram and bigram are used as the baselines for comparison. The graphs of average precisions ($\bar{P}$) vs. average recalls ($\bar{R}$) for all queries of five term types: unigram, bigram, DB, bigrams with unigrams, DB with unigrams for MD1000 and MEDLINE collections are shown in Fig. 5 and 6, respectively.

There are four observations that can be made from results of MD1000 in Fig. 5. For each observation, the explanation and discussions are provided along.

• When consider only the results of the baselines, unigram yields a higher precision than bigram. This can be explained that a unigram is a smaller unit than a bigram, when the unigrams were used to represent the terms, the chance that the unigrams appears in both documents and queries are higher than the bigrams.

• The performance of the DB is slightly better than

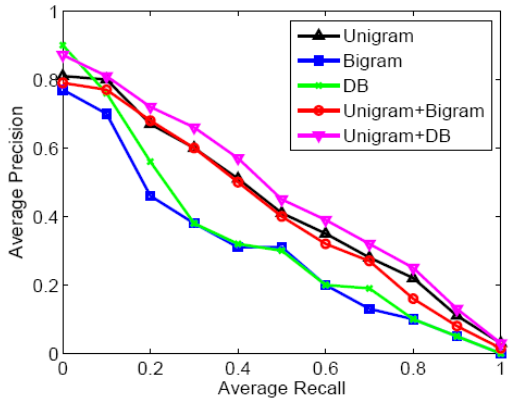

Fig.6: *The interpolated average recall ($\bar{R}$) vs. average precision ($\bar{P}$) of MD1000 for unigram, bigram, DB, bigrams with unigrams, DB with unigrams.*

that of the bigram. The reason is that as the bigram set is a subset of the DB set, the performance of the DB is close to that of the bigram. The DB sets contains more varieties of terms that can increase the chance that DB terms of documents are the same with the queries.

- When DB is combined with unigram, the precision is higher than the conventional unigram at the very early recalls (0.1) and becomes significantly lower than the unigram at the later recalls. This can be explained that the gapped bigrams usually retrieve a very few relevant documents but with high precision (i.e. low recall with high precision) resulting the higher promotion of precision at low recall than at higher recalls.
- DB combined with unigram yields a noticeably higher precision than the plain bigram combined with unigram especially at the early recalls because DB contains more varieties of gapped bigrams, thus it can retrieve relevant documents containing the synonym phrases or sentences caused by adding or removing modifiers from the queries. Consequently, it improves the ranks of the relevant documents because of the cosine similarities added from DB. The highest relevant improvement is 7

For the results of MEDLINE collection shown in Fig. 6, similar observations to those of MD1000 are found such as unigram is better than bigram, DB is better than bigram especially at the early recall, DB with unigram is considerably better than bigram, and DB with unigram is considerably better than bigram with unigram. The highest relative improvement is 12% obtained at recall 0.1 where the precision of DB with unigram is 0.9 and the conventional bigram is 0.8.

5. CONCLUSION

We proposed a new way to improve retrieval using a DB which is a bigram that allows one or more terms to be inserted between its components. The weights of DB depend on its distance from the first component to the second component in the original text. We applied DB with unigram and compared the results with two standard methods: unigram and bigram in the process of transforming the terms into vectors in the vector space model. We tested them with various kinds of terms: unigram, bigram, distance-based bigram, unigram with bigram, and unigram with distance-based bigram on the Thai medical corpus which contains many compound words and writing style expressions. DB combined with the unigram shows better results than unigram and bigram at the low recall. When only some few documents are needed, the new method works well. The relative improvement of DB combined with unigram compared to unigram is up to 12

6. ACKNOWLEDGMENT

The authors would like express our gratitude to the Thailand Research Funds (TRF) for providing us the funding (Project number MRG 5080273)

References

[1] Thomas K. Landauder, P.W.F., Laham, D.: Introduction to latent semantic analysis. Discourses Processes 25 (1998) 259–284

[2] Berry, M.W., Dumais, S.T., O'Brien, G.W.: Using linear algebra for intelligent information retrieval. SIAM Review 37 (1995) 573–595

[3] Pirkola, A., Keskustalo, H., Leppänen, E., Känsälä, A.P., Järvelin, K.: Targeted s-gram matching: a novel n-gram matching technique for crossand mono-lingual form variants. Information Process Management 4 (2002) 231–255

[4] Burkhardt, S., Kärkkäinen, J.: Better filtering with a gapped q-grams. Fundamenta Informaticae 56(1/2) (2003) 51–70

[5] Ian H. Witten, Alistair Moffat, T.C.B.: Managing Gigabytes: Compressing and Indexing Documents and Images. Van Nostrand Reinhold (1994)

[6] Michael Berry, Zhang Drmac, E.R.J.: Matrices, vector spaces, and information retrieval. SIAM Review 41 (1999) 335–862

Pakinee Aimmanee received the B.S. in Mathematics, University of Delaware, Newark, Delaware, USA in 1999. M.S. and Ph.D. in Applied Mathematics, University of Colorado at Boulder, Boulder, Colorado, USA in 2005. Current Position: A lecturer at School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University, Thailand. Research Interest: Information Retrieval, Natural language processing, applied linear algebra

Thanaruk Theeramunkong received the B.S. in Electrical & Electronics Engineering, Tokyo Institute of Technology, Japan M.S. in Computer Science, Tokyo Institute of Technology, Japan Ph.D. in Computer Science, Tokyo Institute of Technology, Japan Academic Rank: Associate Professor Current Position: A lecturer and researcher at School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University, Thailand. Research Interest: Natural language processing, Artificial Intelligence, Knowledge data discovery, Information retrieval, Data mining, Machine learning, and intelligent informatics systems