Using N-gram and Frequent Max Substring Techniques for Index-Term Extraction from Non-Segmented Texts: A Comparison of Two Techniques

Todsanai Chumwatana

Faculty of Information Technology, Rangsit University, Pathumthani 12000, Thailand
Email: todsanai@hotmail.com

ABSTRACT – The amount of electronically stored information in non-segmented texts has grown rapidly and the number of these documents is still increasing. This makes index-term extraction an essential task and some techniques have been proposed for extracting index-terms from non-segmented texts in order to support indexing. In this paper, we investigate two index-term extraction techniques: n-gram and frequent max substring techniques for non-segmented texts. Many research communities have acknowledged that the n-gram technique is one of the viable solutions for extracting index-terms in non-segmented texts such as Chinese, Japanese, Korea, Thai languages and genome or protein in area of bioinformatics. Beside this, the frequent max substring technique has been proposed as an alternative method to extract index-terms. This technique provides significant benefits for indexing non-segmented texts. In this paper, experimental studies and comparison results are shown in order to compare two techniques. From the experimental results, the following observations can be made. The n-gram technique requires less space to extract the index-terms when compare to the frequent max substring technique. Meanwhile, the frequent max substring technique has improved over the n-gram technique in term of performance as it can be applied to many non-segmented texts without the requirement of determining the dimensions of the term.

KEYWORDS – Frequent max substrings, Frequent substrings, n-gram technique, Frequent max substring techniques, n-grams.

1. Introduction and Background

For information retrieval, index-term extraction can be regarded as an important task that has to be performed before constructing the index to allow efficient retrieval. There are a number of techniques have been proposed to extract index-terms for efficient retrieval. However, many challenges are still existing for extracting index-terms from non-segmented texts. Many efforts have been devoted to researching and developing index-term extraction techniques for such problems. While there are many literatures reporting techniques to solve the problems for segmented texts like European languages [1]. Normally, European languages are naturally segmented into individual words that can directly be used as the index-terms and indexed by indexer for efficient retrieval. However, it is known that many Asian languages such as Chinese, Japanese, Korea and Thai are considered as non-segmented texts where the structure of writing is a string of symbols without explicit word boundary delimiters. Words in these languages are not naturally separated by any word delimiting symbols such as white spaces. In addition, there are also other non-segmented texts in area of bioinformatics such as genome or protein sequences. These sequences refer to sequence of characters over specific alphabets which are not based on human nature language. Therefore, extracting the index-terms for these texts becomes a challenging task in the area of information retrieval because they cannot be easily simulated by computer algorithms. Although index-terms can be manually specified by experts, this process is very time consuming and labor intensive. Consequently, some approaches have been proposed in extracting index-terms for non-segmented texts as will be described in the next section.

2. Related Works

Index-term extraction is one of the many techniques that used to automatically extract index-terms from many Asian texts. Previously proposed methods for extracting index-terms in Asian languages can be classified into two main categories: word based and n-gram based approaches. In the word based approach, there are several techniques proposed to split Asian texts such as Chinese [1], Japanese [2], Korea [3] and Thai [4] into term tokens. A word segmentation technique is usually required to extract the index-terms before they can be organized into the index. This technique provides lower accuracy in term of searching due to a segmentation ambiguity [4], [5] when compared with the n-gram technique for non-segmented texts. In addition, most of the word segmentation techniques are language-dependent. They usually rely on language analysis or on the use of dictionary. The preparation of such method is very time consuming. Therefore, word segmentation has become a challenging task in Natural Language Processing (NLP) for Asian texts due to their non-segmented nature. This makes the n-gram technique is more popular technique and widely used to segment many Asian texts due to its being language-independent. The n-gram technique was first introduced and tested as index-terms by Adams in 1991 [6]. This technique is a language-independent approach, which does not require the use of language analysis, or a dictionary or corpus. In information retrieval systems, the n-gram technique is often used for Asian texts where extraction of words is not simple [7], [8]. It can also be applied to other non-segmented texts such as genome or protein sequence [5], [9], [10]. However, determining the dimensions of the gram term is very important issue that should be considered so that they are appropriate for each application. For instance, it has been shown that the bi-gram term is effective for indexing Chinese texts [11], [12], while tri-gram terms are used as index-terms for the protein sequence [13].

Apart from the n-gram technique, the frequent max substring technique is another technique, which has been proposed to extract index-terms from non-segmented texts. This technique was first introduced in 2008 and has been applied in application for indexing for non-segmented texts such as Chinese, Thai languages and genome or protein sequence [14], [15], [16]. It is also successfully applied to many applications in the area of information retrieval [64], [17]. The main strength of this technique is that it was proposed as a language-independent technique, which does not rely on the use of language analysis. This technique also does not require determining the dimensions of the gram term. Because of this, the frequent max substring technique is then applicable for many non-segmented texts such as Chinese and genome sequences, which are regarded as non-segmented texts.

3. Index-Term Extraction Techniques

In this section, we review the details of two techniques for extracting index-terms from non-segmented texts: the n-gram and frequent max substring techniques.

3.1 n-gram Technique

The n-gram technique is used to extract n-grams as index-terms. It has been widely used in information retrieval for non-segmented texts such as Chinese, Japanese, Korea, Thai and genome.

Let us consider a document d as a string of characters $s_1, s_2, ..., s_N$. An n-gram is a substring of n overlap or non-overlap successive characters extracted from texts. Extracting a set of n-grams from the document d can be done by using the 1-sliding technique. That is, sliding a window of length n from s_1 to s_N and storing the characters located in the window. Therefore, the ith n-gram extracted from document d is the substring $s_i, s_{i+1}, ..., s_{i+n}$. Figure 1 shows 2-gram (bi-gram), 3-gram (tri-gram), …, N-gram overlap sequence of the document d containing the string S 'GTCGTCT'.

2-gram	GT, TC, CG, GT, TC, CT
3-gram	GTC, TCG, CGT, GTC, TCT
:	
N-gram	GTCGTCT

Figure 1. The sets of 2-gram, 3-gram, …, N-gram overlap sequence of the document d.

For information retrieval, after non-segmented texts are segmented into serial of index-terms using n-gram technique, all tokenized index-terms are then stored in alphabetical order in the index for fast and efficient retrieval. The index is normally composed of two

elements: the *vocabulary* and *postings file*. The *vocabulary* contains the set of all distinct index-terms that occur in the documents. The *postings file* contains a list of pointers or index-terms positions where they appear in the documents. The following shows the organizing of the index-terms into the index.

From figure 1, we can show an example of posting lists of the 2-gram, 3-gram,…, N-gram which are created on the document *d1* containing the string *S* "GTCGTCT" as shown in figure 2.

d1: GTC G TCT

1 2 3 4 5 6 7

	Vocabulary	***Posting file***
2-gram	GT :	<*d1*, 2, [2, 5]>
	TC :	<*d1*, 2, [3, 6]>
	CG :	<*d1*, 1, [4]>
	CT :	<*d1*, 1, [7]>
3-gram	GTC :	<*d1*, 2, [3, 6]>
	TCG :	<*d1*, 1, [4]>
	CGT :	<*d1*, 1, [5]>
	TCT :	<*d1*, 1, [7]>
:		
N-gram	GTCGTCT:	<*d1*, 1, [7]>

Figure 2. An example of the index of the document containing the string S "GTCGTCT".

To build the index efficiently, the trie data structure is employed to build the index. When constructing the trie data structure, all index-terms are stored by collecting one letter at a time in lexicographical order in the trie data structure. If two or more terms have the same prefix, they will be kept in the same subtree before moving to the next character. Otherwise, if the current character does not match the current nodes in the trie data structure, a new branch will be made to collect the mismatched character, and then moved to the next character. In addition, in any time when the index-terms are kept in a leaf node, the list of term positions is also shown in the leaf node on the trie data structure. Finally, these processes are repeated till the end of the vocabulary.

Figure 3 illustrates the building of an index for 2-gram, 3-gram, …., N-gram using these processes as shown in figure 3.

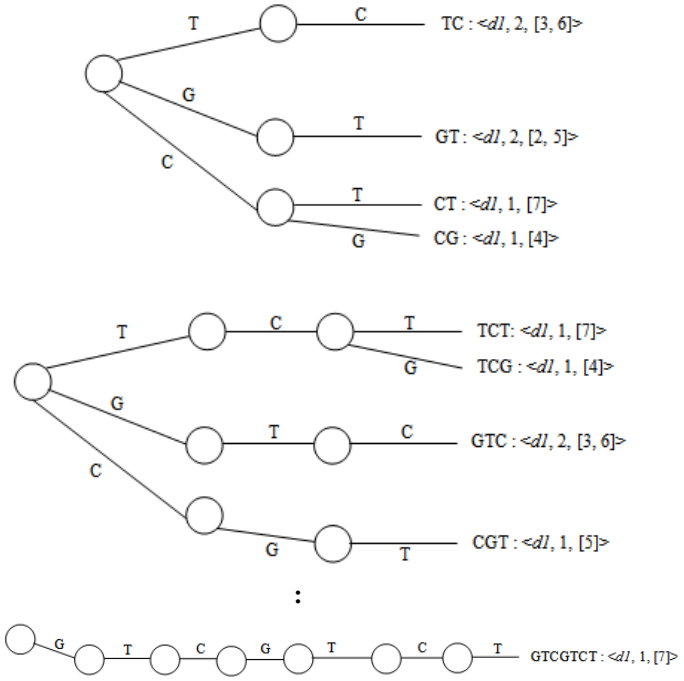

Figure 3. The 2-gram, 3-gram, …., N-gram indexes.

The advantages of the n-gram technique are the error-tolerance and language-independence [18], [19], [20]. Due to these advantages, this method is one of the promising index-term extraction techniques that is acknowledged by many Asian research communities. It is also used widely in genome or protein sequences for searching a string since this technique does not rely on the dictionary or language analysis.

However, as mentioned, the parameters for using the n-gram technique should be adjusted so that they are appropriate for each application. In information retrieval systems, the n-grams is often used as a index-term for Asian languages such as Korean, Chinese, and Japanese, where extraction of words is not simple [21] and the parameter for n-gram is regarded as an important decision factor in performing indexing for these languages. For instance, survey shows that the bi-gram index-term has given significant retrieval effectiveness for Chinese texts since most Chinese words can comprise two characters. Furthermore, most Chinese bi-gram index-terms do not lose the semantics of words. Chinese characters can also be linked together to make a phrase. A phrase can consist of two, three or more characters, but there are no spaces between Chinese characters except punctuation marks

such as , or 。(full stop). In Japanese, *n* can also be equal to two characters which is also suitable for n-gram index because most Japanese words are ideographs that are composed of one or two characters [22]. For instance, let a simple Chinese text document *d* containing the string '假的作真的时真的亦为假的' and this document *d* is a string of characters s_1, s_2, ..., s_N. Therefore, the i^{th} bi-gram term extracted from the document *d* is the substring s_i, s_{i+1}, ..., s_{i+n}. Figure 4 shows the bi-gram index-terms overlap sequence of the document *d* containing the string *S* '假的作真的时真的亦为假的'.

Let *d*:	假的作真的时真的亦为假的
Bi-gram index-terms:	假的, 的作, 作真, 真的, 的时, 时真, 真的, 的亦, 亦为, 为假, 假的

Figure 4. Example of bi-gram index-terms from document.

Meanwhile, Jaruskulchai [23] showed that the more probable *n* for Thai language, that is not ideographic, should be greater than two, to achieve the retrieval effectiveness since the combination of some consonants and vowels are more defined words. The top 20 of the high frequencies' 3-gram and 4-gram are more complete words in the Thai language. In bioinformatics, CAFE [24] is a well known method which uses the n-gram index. It uses 9-grams for the genome sequence and 3-grams for the protein sequence as index-terms.

Due to the above reasons, we cannot specify the best parameter for n-gram index-terms of non-segmented texts because of the variety of data. In order to support every parameter for n-gram approach, the frequent max substring technique has been proposed to extract all frequent index-terms for every parameter as will be described in the next section.

3.2 Frequent Max Substring Technique

In order to support every parameter for n-gram index-terms, the frequent max substring technique is proposed to generate frequent long substring as the index-terms [25]. Frequent max substring technique is based on text mining that describes a process of discovering useful information or knowledge from unstructured texts. This technique is used to classify index-terms called frequent max substrings from the non-segmented texts where the word boundaries are not clearly defined. The frequent max substrings refer to the substrings that appear frequently (at a predetermined frequency *f*) and have the maximum length of n-grams on the given string, so these terms are likely to be the patterns of interest. The set of frequent max substrings is also able to contain all frequent substrings which appear on the given texts. We extract the set of frequent max substrings by using the frequent max substring technique. This technique uses Frequent Suffix Trie or FST data structure to explore the index-terms [15]. The FST data structure is similar to suffix trie structure [26] that is an efficient substring enumeration method. However, FST data structure enumerates substrings with their frequencies and positions information while suffix trie structure enumerates only substrings without their frequencies information. Therefore, we employ FST data structure in order to support extracting frequent max substrings. In the frequent max substring technique, the parameter and the predetermined frequency are applied to reduce the number of the index-terms. This method uses the two reduction rules: 1) reduction rule using the predetermined frequency to check extracting termination, 2) reduction rule using superstring definition to reduce the number of index-terms extracted. This technique also uses heap data structure to support computation [15]. As a result, this technique extracts only the frequent long n-grams, called frequent max substrings, as index terms which contain all frequent n-gram index terms from the text in order to improve the performance of searching for information retrieval.

In order to explain the concept, the following briefly describes the steps of extracting frequent max substrings as index-terms from genome sequencing.

Let genome sequence *S* = 'ATGATGT'

and predetermined frequency *f* = 2

1. We extract 1-gram, 2-gram index-terms and so on with their frequencies and positions. Only index-terms that occur at least at the predetermined frequency *f* will be extracted, and are sorted in order of occurring in the texts on the FST data structure for further processes.

2. Then, we extract the frequent max substrings by selecting index-terms having no super-substrings from the set of the frequent index-terms, also called frequent substrings, in order to reduce the number of the index-terms.

To illustrate the applicability of the frequent max substring technique for genome sequencing, the following example shows the process of the frequent max substring technique using Min Heap and two reduction rules to extract the frequent max substrings from genome sequencing

Let genome sequence *S* = 'ATGATGT'

Position(.pos) = 1 2 34 567

and predetermined frequency *f* = 2

Min-heap structure

Firstly, all substrings with a length of 1 are extracted, together with their frequencies and list of positions. The frequencies of these substrings are then checked in order to select only the frequent substrings with a length of 1. These frequent substrings are finally kept in the min-heap structure for further processes.

A, 2 .pos=1, 4	T, 3 .pos=2, 5, 7	G, 2 .pos=3, 6

Next, <A, 2> is removed from min-heap in order to indicate that <A, 2> is detected and extracts its child substrings for the next process. After <A, 2> is removed from min-heap, the algorithm extracts child substrings of <A, 2> using list of positions or pointers of <A, 2> to reduce time complexity. Child substrings consist of <AT, 2>. <AT, 2> is kept in min-heap using the insertion rule, because <AT, 2> is the substring that occurs in two different positions in string s.

T, 3 .pos=2, 5, 7	AT, 2 .pos=2, 5	G, 2 .pos =3, 6

<T, 3> is removed from min-heap, after which child substrings of <T, 3> are extracted using the list of positions or pointers of <T, 3>. Child substrings consisting of <TG, 2> and <T$, 1>. <G, 2> are deleted from min-heap because <TG, 2> is a proper superstring of <G, 2> at the same frequency, and <TG, 2> is kept in min-heap instead, using the insertion rule, because its frequency is equal to the predetermined frequency.

AT, 2 .pos=2, 5	TG, 2 .pos =3, 6

<AT, 2> is removed from min-heap and then its child substrings are extracted using its list of positions (pointers). They consist of <ATG, 2>. <TG, 2> is deleted from min-heap because <TG, 2> is a substring of <ATG, 2> with the same frequency. After that, <ATG, 2> is kept in min-heap using the insertion rule because <ATG, 2> is the substring that occurs in two different locations in string s.

ATG:2 .pos =3, 6	

<ATG, 2> is removed from min-heap and then its child substrings are extracted using its list of positions. They consist of <ATGA, 1> and <ATGT, 1>. They are not kept in min-heap because their frequencies are less than predetermined frequency.

The algorithm will stop when min-heap is empty. This means all substrings in min-heap were detected and processed completely.

From the above process, the resulting frequent suffix trie or FST structure can be depicted in Figure 5.

Figure 5 shows the FST structure using the frequent max substring technique. The set of frequent max substrings is {<s, 2>, <i, 3>, <ive, 2>}.

To illustrate the applicability of the frequent max substring technique for other non-segmented texts, we show another example of the FST structure that was constructed on the Chinese text as shown in following

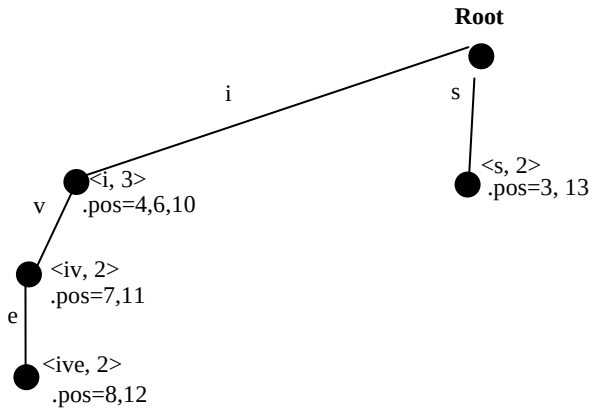

Figure 5. Frequent suffix trie structure using frequent max substring technique

Let Chinese text *S* = '假的作真的时真的亦为假的' and predetermined frequency *f* = 2

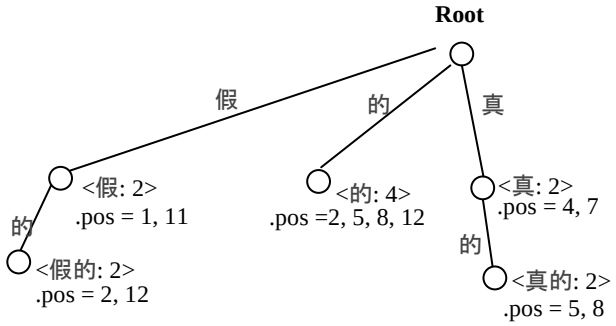

Figure 6. The FST data structure using the frequent max substring technique on the Chinese text.

Figure 6 shows the FST structure. The set of frequent max substrings is {<的, 4>, <假的, 2>, <真的, 2> }

4. Experiments and Comparison Results

In this section, the experiment and comparison result of extracting index-terms from Chinese texts using the bi-gram and frequent max substring techniques are presented. Bi-gram is chosen because it is one of the most widely used techniques for indexing Chinese texts [11]. In this experiment, the predetermined frequency *f* is set to 2, therefore only index-terms that occur at least two times are of interest. The text collection used for evaluation is a set of Chinese texts obtained from the website: http://www.personal.umich.edu/~dporter/sampler/sampler.html.

The texts have varying lengths. The set of texts consists of 20 texts and contains 28,522 characters. The text lengths range from 564 to 2,187 characters. The n-gram and frequent max substring techniques are used to extract the set of index-terms from the set of Chinese texts, at the predetermined frequency *f* = 2.

In order to compare the two index-term extraction techniques: the bi-gram and the frequent max substring techniques for Chinese texts, the number of index-terms extracted from both techniques is compared. In Figure 7, a comparison of two techniques is presented. The vertical axis represents the number of index-terms and the horizontal axis represents the text size (*n* characters).

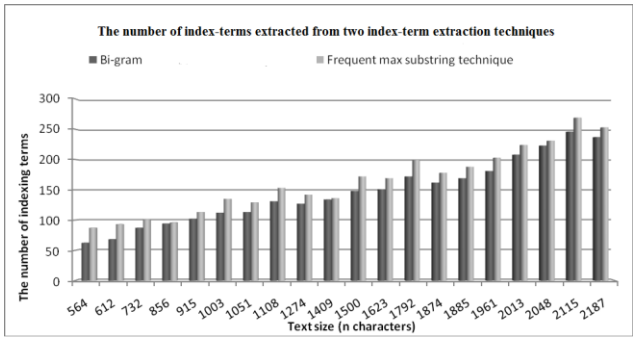

Figure 7. Comparison of number of index-terms extracted from bi-gram and frequent max substring techniques

From the comparison results, it can be observed that the bi-gram technique extracted a slight lesser number of index-terms than the frequent max substring technique. The index-terms extracted from the bi-gram technique are small and each term consists of two characters. Meanwhile, the frequent max substring technique generates slightly more index-terms compared to the bi-gram technique. The index-terms extracted from the frequent max substring technique have varying lengths. This means more space is needed to store index-terms compared to the bi-gram technique.

In addition, the experiment and comparison results of extracting index-terms from other sorts of non-segmented texts such as genome sequence and Thai language were also discussed in this paper. We also performed the use of the n-gram and frequent max substring techniques for the Thai language and genome sequences. For the genome sequence, 9-gram is chosen because 9 is the best dimensions to represent the genome sequence [24]. In order to compare the two different algorithms for genome sequencing, the number of index-terms that are enumerated by using 9-gram technique and the number of index-terms that are enumerated by using the frequent max substring technique are compared. The dataset used for the evaluation is the genome sequence found on the website: http://www.broadinstitute.org/cgi-bin/annotation/methanosarcina/download-sequence.cgi.

Consequently, it can be observed that the experiment results of extracting index-terms from the genome sequence are similar to the experiment results of extracting index-terms from the Chinese language. In case of Thai language, 3 is chosen as a dimensions of the term for the n-gram technique. This is because the more probable *n* for Thai language, that is not ideographic, should be greater than two, to achieve the retrieval effectiveness since the combination of some consonants and vowels are more defined words. The top 20 of the high frequencies' 3-gram is more complete words in the Thai language. Again, the number of index-terms that are extracted by using 3-gram technique and the number of index-terms that are extracted by using the frequent max substring technique from the Thai language are compared. The results have shown that the frequent max substring technique provides a better result in term of performance when compared to the 3-gram technique. This is because the index-terms extracted from the frequent max substring technique have varying lengths meanwhile each index-term extracted from the 3-gram technique consists of 3 characters. This makes the result from the frequent max substring technique is more applicable than the 3-gram technique as the words in Thai language naturally have varying lengths, and they may consist of two, three or more characters which cannot specify the certain number of characters per word. Therefore, the 3-gram technique then cannot provide such a good performance for Thai language. In contrast, the frequent max substring technique shows the beneficial results because most index-terms from this technique are more complete words or sentences in Thai language.

Furthermore, the main advantage of the frequent max substring technique is that it is applicable for any types of non-segmented texts without the requirement of determining the dimensions of the term. This shows that the frequent max substring technique is a versatile index-term extraction technique. For instance, the frequent max substring technique extracts any length of index-terms as long as those index-terms occur at least at the predetermined frequency *f*. Therefore, it can directly be applied to Chinese, Japanese, Korea, Thai languages and genome or protein sequence etc, without specifying the dimensions.

5. Conclusion

This paper describes and compares two index-term extraction techniques: the n-gram and frequent max substring techniques for non-segmented texts. These two techniques are regarded as a viable solution for extracting index-terms in non-segmented texts and also used in the area of bioinformatics. From the experimental studies and comparison results, we have observed that the frequent max substring technique requires slightly more space to extract the index-terms when compared to n-gram technique. However, the frequent max substring technique provides the significant benefit in term of versatileness. The frequent max substring is applicable for any types of non-segmented texts and can directly be applied to non-segmented texts without the requirement of determining the dimensions of the term. This shows that the frequent max substring has improved over the n-gram technique in term of performance, meanwhile the n-gram technique requires less memory for extracting index-term.

References

[1] K. L. KWOK, 1997 Comparing representations in Chinese information retrieval, In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Philadelphia, Philadelphia, pp. 34-41.

[2] F. H. a. W. B. Croft, 1993 A Comparison of Indexing Techniques for Japanese Text Retrieval, In *Proceedings of ACM SIGIR International Conference on Research and Development in Information Retrieval*, pp. 237-246.

[3] J. H. L. a. J. S. Ahn, 1996 Using n-Grams for Korean Text Retrieval, In *Proc. Int'l Conf. on Information Retrieval, ACM SIGIR*, Zurich, Switzerland, pp. 216-224.

[4] R. S. a. D. Smith, 2001 Information Extraction for Thai Documents, *International Journal of Computer Processing of Oriental Languages (IJCPOL),* pp. 14(2):153-172.

[5] G. Navarro, 2001 A Guided Tour to Approximate String Matching, In *ACM Computing Surveys*, pp. 31-88.

[6] E. Adams, "*A Study of Trigrams and Their Feasibility as Index Terms in a Full Text Information Retrieval System*." PhD thesis, George Washington University, USA, 1991.

[7] M. M. P Majumder, B.B. Chaudhuri, 2002 N-gram: a language independent approach to IR and NLP, In *International conference on Universal Knowledge*.

[8] W. A. T. CAVNAR, J, 1994 N-gram based text categorization, In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval* (Las Vegas, NV, 1994), pp. 161-175.

[9] K. Y. W. M. S. Kim, J. G. Lee, and M. J. Lee, 2005 n-Gram/2L: A Space and Time Efficient Two-Level n-Gram Inverted Index Structure, In *VLDB, Trondheim*, Norway, pp. 325-336.

[10] H. E. Williams, 2003 Genomic Information Retrieval, In *Proc. the 14th Australasian Database Conferences*.

[11] L. F. Chien, "Fast and Quasi-Natural Language Search for Gigabytes of Chinese Texts," in *Proceedings of 18th ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, USA, 1995, pp. 112-120.

[12] T. Liang, S. Y. Lee, and W. P. Yang, "Optimal Weight Assignment for a Chinese Signature File," in *Journal of Information Processing and Management*, vol. 32, no. 2, pp. 227-237.

[13] H. E. Williams and J. Zobel, "Indexing and Retrieval for Genomic Databases," in *IEEE Transaction on Knowledge and Data Engineering*, 2002, pp. 63-78.

[14] T. Chumwatana, K. W. Wong, and H. Xie, "An Automatic Indexing Technique for Thai Texts using Frequent Max Substring," in Eighth International Symposium on Natural Language Processing, 2009 (SNLP '09) Bangkok, Thailand, 2009.

[15] T. Chumwatana, K. W. Wong, and H. Xie "Frequent Max Substring Mining for Indexing," International Journal of Computer Science and System Analysis (IJCSSA), India, 2008.

[16] T. Chumwatana, K. W. Wong, and H. Xie, "Using Frequent Max Substring Technique for Thai Keyword Extraction used in Thai Text Mining," in *2nd International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT 2010)*, Bali, Indonesia, 1-2 July 2010.

[17] T. Chumwatana, K. W. Wong and H. Xie, 'Non-segmented Document Clustering Using Self-organizing map and Frequent Max Substring Technique', *In16th International Conference on Neural Information Processing (ICONIP 2009)*, Bangkok, Thailand, 2009.

[18] R. B.-Y. a. B. Ribeiro-Neto, 1999 *Modern Information Retrieval*: ACM Press.

[19] J. M. a. P. McNamee, 2003 Single N-gram Stemming, In *Proc. Int'l Conf. on Information Retrieval, ACM SIGIR*, Toronto, Canada, pp. 415-416.

[20] D. S. Ethan Miller, Junli Liu, and Charles Nicholas 2000 Performance and Scalability of a Large-Scale N-gram Based Information Retrieval System, *Journal of Digital Information*, vol. 1, No. 5, pp. 1-25.

[21] O. Y. a. M. Toru, 1998 Optimizing query evaluation in n-gram indexing, In *Proc. Int'l Conf. on Information Retrieval, ACM SIGIR*, Melbourne, Australia, pp. 367-368.

[22] B. A. Ogawa Yasushi, and Iwasaki Masajirou, 1993 A New Indexing and Text Ranking Method for Japanese Text Databases Using Simple-Word Compounds as Keywords, Database Systems for Advanced Applications'93, In *Proc. of the Third International Symposium on Database System for Advanced Applications*, pp. 197-204.

[23] V. Sornlertlamvanich, Word Segmentation for Thai in Machine Translation System, Bangkok.

[24] H. E. W. a. J. Zobel, 2002 Indexing and Retrieval for Genomic Databases, In *IEEE Trans. on Knowledge and Data Engineering*, pp. pp. 63-78.

[25] T. Chumwatana, Kok Wai Wong and Hong Xie, 2008 Thai Text Mining to Support Web Search for E-commerce, In *The 7th International Conference on e-Business (INCEB2008)*, Bangkok, Thailand.

[26] D. Gusfield, 1997 *Algorithms on Strings, Trees and Sequences Computer Science and Computational Biology*. Cambridge: Cambridge University Press