KMITL Sci. J. Vol. 5 No. 1 Feb. 2005

THAI TEXT TRANSFORMATION FOR COMPRESSION

K. Sermkawinrak, S.Intakosum, and V.Boonjing

Software Systems Engineering Laboratory
Department of Mathematics and Computer Science
Faculty of Science, King Mongkut's Institute of Technology at Ladkrabang (KMITL)
Ladkrabang Bangkok 10520, THAILAND

ABSTRACT

The paper presents a new Thai-text transform algorithm to enhance compression using the list of frequently used Thai words/phrases. The approach is to increase redundancy in text by encoding it into intermediate form. The encoding scheme uses the list of fixed length codes for frequently used Thai words/phrases to substitute words/phrases in text with their codes. Algorithm performance is measured in terms of compression ratio. There are three major implementations for experiment. The first is to include all 511 frequently used Thai words/phrases. Therefore, a three-byte code is assigned to each word/phrase. The second uses a two-byte code because it concerns with the first 255 most frequently used words/phrases. The last concerns the first 109 most frequently used words/phrases with one-byte code for each word/phrase. An experiment was made using each text and its transformed version as input to standard compression programs. The result shows that the transformed text gives compression ratio significantly better than its original one.

1. INTRODUCTION

In today digital age, a lot of information is in digital form and is stored in various storage media that can be accessed easily anytime and anywhere. As a result, a large data-storage space is needed. Therefore, many researches have been carried out with an objective to reduce data size. However, most of the researches emphasized on the development and improvement of data compression algorithms. As a dramatically increasing of information, the algorithm development solely may not be enough. Thus it brings about a new approach to enhance compression ratio that is called data transformation or transformation for short. The main purpose of transformation is to convert the original data into intermediate form that can increase data compression ratio, for example, increase the redundancy of text in a text file. Some researches such as [1], [2], and [4], present transformation algorithms the can be applied to English texts. The results from those researches are shown that the transformation effectively improves the data compression ratio. However, those algorithms can be applied only to English language text. This paper, therefore, proposes a new Thai text transformation approach for data compression. It starts from a brief overview of English text transformation algorithms, then the proposed Thai text transformation algorithm, and ends with conclusion and recommendation.

2. ENGLISH TEXT TRANSFORMATION ALGORITHMS

General transformation procedure consists of two parts: 1) encoding, and 2) decoding. Encoding part is used to convert an original data into intermediate form that can enhance data compression ratio, while decoding is to reform the encoded data back into its original form. Some well-known algorithms are:

2.1 Burrows-Wheeler Transformation (BWT)

The BWT method [1],[2] reorders data bytes in the original file so that data bytes with the same value are adjacent. As a consequence, the redundancy is increased. This helps improvement of the data compression ratio for some data compression programs employing this algorithm such as Run Length Encoding (RLE) and Huffman coding. The advantage of this method is that it can be applied to any data types. However, it requires a large number of system resources for transforming a large data. [3]

2.2 Length Index Preserving Transformation (LIPT)

LIPT [4] requires a static dictionary, a computer that needs to use this transformation must have the same dictionary. This technique starts by reading a word in the source file; compare it with the words stored in the dictionary. Once the source word is found, it will be encoded to represent the position and the length the word in dictionary; the encoded word therefore will replace the original word in the source file. In the decoding process the source file is read and the encoded word will be transformed back to its original by consulting the dictionary.

LIPT always reduced the size of the original file, therefore, the size of the target-compressed file should be also decreased. However, it requires static dictionary, then, lead to three-limitation [5] :

- The wastes of space since some stored words in dictionary have never been used.
- The difficulties in managing and modifying dictionary because the size of dictionary is too large.
- The transformation cannot be done if the word is not in the dictionary.

2.3 Semi-Dynamic Length Index Preserving Transformation and Dynamic Length Index Preserving Transformation (SDLIPT/DLIPT)

Dissunrat [5] proposes various transformation techniques that breakthrough some limitations of LIPT technique. These techniques are Semi-Dynamic Length Index Preserving Transformation (SDLIPT) and Dynamic Length Index Preserving Transformation (DLIPT).

Both techniques are based on the LIPT technique. The SDLIPT uses static dictionary, in the same way as the LIPT, and uses dynamic dictionary for storing new words found from the source text. The DLIPT proposes to use only dynamic dictionary. The dictionary is created and updated at the time of transformation processing.

1. THE THAI TEXT TRANSFORMATION ALGORITHM

From the study of the algorithms in section 2, the authors found that BTW is not proper to use for Thai text transformation for two major reasons, it requires a lot of space, and does not work well with in the context of Thai language. LIPT, SLIPT and DLIPT are the better choices. However, among those three algorithms, although SLIPT, and DLIPT are more general but they require more complex ways to handle the dictionary. In addition, based on the research on Thai frequently used words in [6], and [7], the dynamic dictionary are not required, therefore, the Thai text transformation algorithm presented in this paper is based on the concept of LIPT.

The proposed transformation algorithm converts the original data into an intermediate form that can increase the redundancy in text, and reduce data size through encoded text. As an experiment there are three implementations for this algorithm. The first one is to store all of the words from [6], and [7] into dictionary (see 3.1). The second and the third implementations use the first 255, and 109 frequently found words respectively.

3.1 Dictionary of Thai Words

Before transforming Thai text, the computer that is used for encoding/decoding Thai text transformation must have the dictionary of Thai frequently used words. The words in dictionary are derived from the studies of Thai text analysis [6],and [7]. The data structure for this dictionary is as follows:

[Encoded word][Original Thai word]

Each encoded word is used to refer to Thai word in the dictionary and also to separate among words in data stream. The implementations of Thai text transformation are developed based on the number of bytes of the encoded words those are two-byte code and one-byte code dictionary respectively.

3.2 Transformation using Two-Byte Code dictionary

1) Thai Text transformation using all 511 Thai frequently used words

In this implementation, all 511 Thai frequently used words have been stored in dictionary. According to the study on Thai word analysis [6], the probability of having all 511 words in the text is around 80 % and the average length of all words in dictionary is 3.6 characters. In this scheme the replacement of the original word is by three-byte word. Two bytes comes from encoded word from dictionary, and one byte is used to store the symbol to indicate that the word is an encoded word, the '*' symbol is used for this implementation. Therefore, the size of each replaced word is, in average, 0.6 characters decreased.

2) Thai text transformation using the first 255 most frequently used words

In this implementation, the first byte of the encoded word in dictionary is not used for encoding purpose, only the second byte is used for encoding the first 255 frequently used words. Therefore, the original word is replaced by two-byte word, one byte is '*', and the other is an encoded byte from the dictionary. Based on [6] the probability of having all 255 words in the source text is 66% and the average length of a word is 3.4 characters. Since this approach uses 2 bytes of an encoded word, thus, the size of each encoded Thai word is decreased by 1.4 characters in average.

3. 3 Transformation using One-Byte Code dictionary

In this implementation, the size of the encoded word is reduced to one byte. The symbol '*' is no longer used as a starting point of the encoded word. Instead, all ASCII code for English characters and some data communications code are used for encoding since Thai characters do not use the ASCII code in that range. This experiment used the first 109 most frequently used words. According to the study on Thai word analysis [6], the probability of having all 109 words in the text is around 50 % and the average length of word is 3.2 characters. Since only one byte is used for an encoded word, therefore the size of each word is reduced by 2.2 characters in average.

3.4 Comparison of the three implementations

The characteristics of the above implementations are summarized as shown in Table 1.

Table 1. Characteristics of three implementations of Thai Text Transformation Algorithm

Thai Text Transformation	Two-Byte Code Transformation		One-Byte Code Transformation
	All 511 frequently used words/phrases	First 255 most frequently used words/phrases	
Words/Phrases in dictionary	511 Words/Phrases	255 Words/Phrases	109 Words/Phrases
Bytes used for one encoded word	3 Bytes	2 Bytes	1 Bytes
Adding Words/Phrases into dictionary	Enable	Unable	Unable
Increasing data size in average	0.6 Byte	1.4 Bytes	2.2 Bytes

The benefit of the first implementation is that the whole frequently used words can be used, in addition, the dictionary has spaces left to store more words. The disadvantage, when comparing to the next two implementations, is that the size of the transformed text is larger since it requires more bytes for the encoded text. For the second, and the third implementations they give less file size, however, there may be a chance that some text files may contain most of the words that are frequently used word but are not stored in the dictionary, in this case, the file size may not be reduced. Moreover, in the first and second implementations since the special symbol '*' is used in encoded word, if more words in the original file need to be replaced this means that many '*' symbols will be occurred in the text. Data compression programs may consider these symbols as redundant data, this may result in better data compression ratio. The third implementation, however, does not benefit from this since no special symbol is needed. However, the results (see section 4) show that the text files that use these implementations gain better compression ratio than the same text files that apply data compression program directly.

4. RESULTS

In order to test the efficiency of Thai Text Transformation algorithm, the authors implemented three prototype programs based on the implementation techniques stated in section 3. A hundred of general Thai texts are used as test samples. Some well-known data compression programs such as, Huffman coding, Arithmetic coding, PKZIP, ARJ and BZIP2 are used to compressed data. The purpose of the test is to find average percent of reduction on the file size after compression without, and with three implementations of Thai text transformation algorithm. The results can be summarized as shown in Table 2.

Table 2. Average percent of reduction on file size without/with Thai text transformation

Transformation Compression	None	TTT3*	TTT2*	TTT1*
None	-	4.2646	19.8324	24.9525
Huffman	28.6452	27.2346	33.3672	35.0974
Arithmetic	31.3343	33.3682	39.123	40.0621
PKZIP	54.1294	54.8231	57.0019	58.389
ARJ	54.0691	54.905	57.0008	58.3892
BZIP2	57.9305	60.1681	59.705	59.1513

For more details on this test see [10].

Note:*

TTT3 represents "Thai Text transform using all 511 Thai frequently used words"
TTT2 represents "Thai text transform using the first 255 most frequently used words"
TTT1 represents "One-byte code transformation"

5. CONCLUSIONS AND RECOMMENDATIONS

The improvement of data compression algorithms alone may not be enough to gain the higher data compression ratio. One way to solve the problem is to perform data transformation prior to data compression. The basic idea of data transformation is to convert the original data into an intermediate form that decreases the size of the original data and/or increase redundancy in the original text. Many previous researches show the successful of the data transformation to English text files. This paper, therefore, proposes a new Thai text transformation algorithm based on data transformation and Thai word analysis researches. The major idea is to create a dictionary that contains frequently found Thai words in general Thai text, and used this dictionary as a reference to convert the original data into an appropriate intermediate form for Thai word. Three implementations of this algorithm are developed based on the numbers of frequently used Thai words stored in dictionary, all 511 words, 255 words, and 109 words respectively. The numbers of words used in dictionary result in the numbers of bytes used to encoding words. This may affect the data compression ratio in the sense that the fewer bytes used to encoding word may lead to the reduction on the size of the compressed text. On the other hand, one-byte encoding word may not increase redundancy on the original text that may lead to the higher data compression ratio. The test is performed on one hundred samples of general Thai text. Some well-known data compressions programs are used. The results show that the average data compression ratio of set of text files that used Thai text transformation is higher than same set of text files that use only data compression program.

Although the results show the successful of the proposed Thai text transformation algorithm, the following researches may be helpful:

1) The set of frequently used Thai words that is used to form dictionary is based on research conducted in 1984. The same research may need to be redone again to update some changes, which may happen.
2) The size of the encoded word affects the size of the transformed data as well as the size of the compressed data. The research should be done to find the conclusion that how many percents of frequently used words should be store into dictionary.
3) The proposed Thai text transformation algorithm concerns solely on using Thai dictionary to transform only Thai text. The development of multi-language dictionary would be helpful in order to apply this algorithm for multi-language documents. The result from the previous suggestion can be helpful to maintain the acceptable size of the dictionary.

REFERENCES

[1] Burrows M., Wheeler D.J. **1994** "A Block-Sorting Lossless Data Compression Algorithm. SRC Research Report 124, Digital Systems Research Center, Palo Alto, CA.

[2] www. http://www.arturocampos.com/ac_bwt.html

[3] Lerwongrat S. **1997** Text Compression by Sorting Transformation. M.S. Thesis in Computer Science, Faculty of Graduate Studies, Mahidol University.

[4] Awan F. and Mukherjee A. **2001** LIPT: A Lossless Text Transform to improve compression. Proceedings of International Conference on Information and Theory, Coding and Computing, IEEE Computer Society, Las Vegas, Nevada.

[5] Dissunrat K. **2001** Text Compression with Modified Length Index Preserving Transformation Using Semi-Dynamic and Dynamic Dictionary. M.S. Thesis in Computer Science, Faculty of Graduate Studies, Mahidol University.

[6] Poovarawan Y. **1984** Thai Word Analysis. (in Thai language) Microcomputer Res. Lab., Computer Engineering, Faculty of Engineering, Kasetsart University.

[7] Poovarawan Y., Imarom W. **1986** Thai Syllable Separater by Dictionary. (in Thai language) Microcomputer Res. Lab., Computer Engineering, Faculty of Engineering, Kasetsart University.

[8] Poovarawan Y., Keretho S. **1983** Suggestion for Thai Standard Character Code. (in Thai language) Microcomputer Res. Lab., Computer Engineering, Faculty of Engineering, Kasetsart University.

[9] Poovarawan Y., Wongchaisuwat C. **1989** Design and Compression of Thai Words in Dictionary for Spelling Check (in Thai language) Microcomputer Res. Lab., Computer Engineering, Faculty of Engineering, Kasetsart University.

[10] Sermkawinrak K. **2005** Thai Text Transformation for Data Compression (in Thai language). M.S. Thesis in Computer Science, School of Graduate Studies, King Mongkut's Institute of Technology Ladkrabang.