การแปลงข้อความภาษาไทยเพื่อการบีบอัดข้อมูล

THAI TEXT TRANSFORMATION FOR DATA COMPRESSION

คัมภีร์ เสริมกวินรักษ์

KAMPEE SERMKAWINRAK

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ.2548

ISBN 974-15-1363-1

THAI TEXT TRANSFORMATION FOR DATA COMPRESSION

KAMPEE SERMKAWINRAK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN COMPUTER SCIENCE

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2005

ISBN 974-15-1363-1

หัวข้อวิทยานิพนธ์	การแปลงข้อความภาษาไทยเพื่อการบีบอัดข้อมูล
นักศึกษา	นายคัมภีร์ เสริมกวินรักษ์
รหัสประจำตัว	45064603
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์
พ.ศ.	2548
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร.ศรัณย์ อินทโกสุม

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนาขั้นตอนวิธีใหม่สำหรับการแปลงข้อความภาษาไทย โดยใช้รายการของคำ/วลีที่พบบ่อยในภาษาไทย เพื่อช่วยเพิ่มประสิทธิภาพการบีบอัดข้อมูล แนวคิดที่ใช้คือการเพิ่มความซ้ำซ้อนในข้อความ โดยการเข้ารหัสให้อยู่ในรูปแบบระหว่างกลาง ในการเข้ารหัสจะใช้รายการของรหัสที่มีความยาวตรึง สำหรับคำ/วลีในภาษาไทยที่ใช้บ่อย ไปแทนที่คำ/วลีที่พบในเอกสาร การวัดประสิทธิภาพของขั้นตอนวิธีจะวัดจากอัตราการบีบอัดข้อมูล ในการทดสอบผู้วิจัยได้พัฒนาโปรแกรมขึ้นทั้งหมด 3 โปรแกรม โดยโปรแกรมแรกจะใช้คำที่ใช้บ่อยทั้งหมดซึ่งคือ 511 คำ ดังนั้นจะต้องใช้รหัสขนาด 3 ไบต์สำหรับคำ/วลีแต่ละตัว โปรแกรมที่สองใช้รหัสขนาด 2 ไบต์ เนื่องจากจะใช้คำที่พบบ่อย 255 คำแรกสำหรับคำ/วลีแต่ละตัว และโปรแกรมสุดท้ายจะใช้คำที่พบบ่อย 109 คำแรก จึงใช้รหัสขนาด 1 ไบต์สำหรับคำ/วลีแต่ละตัว การทดลองทำโดยใช้โปรแกรมบีบอัดข้อมูลมาตรฐานกับข้อความที่ยังไม่ได้แปลง และข้อความที่ผ่านการแปลงแล้ว ผลลัพธ์จากการทดลองปรากฏว่าข้อความที่ผ่านการแปลงข้อความ จะมีอัตราการบีบอัดข้อมูลที่ดีกว่าข้อความต้นฉบับอย่างมีนัยสำคัญ

Thesis Title	Thai Text Transformation for Data Compression
Student	Mr. Kampee Sermkawinrak
Student ID	45064603
Degree	Master of Science
Programme	Computer Science
Year	2005
Thesis Advisor	Asst.Prof.Dr.Sarun Intakosum

ABSTRACT

The objective of this research is to develop a new Thai-text transform algorithm to enhance compression using the list of frequent 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 frequent 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 frequent 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 frequent used words/phrases. The last concerns the first 109 most frequent used words/phrases with one-byte code for each word/phrase. An experiment is 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.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ด้วยคำแนะนำและความช่วยเหลืออย่างดียิ่งของ ผศ.ดร.ศรัณย์ อินทโกสุม ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณ รศ.ดร.วีระ บุญจริง ที่คอยตักเตือนและให้คำแนะนำในการทำวิทยานิพนธ์อย่างดียิ่งมาโดยตลอด

ขอขอบคุณมูลนิธิเพื่อการศึกษาคอมพิวเตอร์และการสื่อสารที่ให้ทุนสนับสนุนการทำวิจัยในครั้งนี้

ขอขอบคุณพี่นคร สาแก้ว ที่ให้คำแนะนำและช่วยเหลือทางด้านการเขียนโปรแกรมตลอดจนให้คำปรึกษาเรื่องทั่วๆไป

ขอขอบคุณเพื่อนๆ นักศึกษาทุกคนที่ช่วยเหลือให้คำแนะนำต่างๆ และยังให้กำลังใจต่อผู้วิจัยอย่างใกล้ชิดตลอดมา

สุดท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณบิดามารดาที่ได้ให้กำลังใจแก่ผู้วิจัยเสมอมา

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

คัมภีร์ เสริมกวินรักษ์

สารบัญ

สารบัญ(ต่อ)

หน้า

สารบัญ(ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

สารบัญรูป(ต่อ)

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ข้อมูลข่าวสารมีความสำคัญและมีบทบาทในชีวิตประจำวัน ทั้งทางด้านธุรกิจ ด้านเกษตรกรรม ด้านอุตสาหกรรม ด้านกฎหมาย ด้านวิทยาการแพทย์ ฯลฯ ทำให้เกิดความจำเป็นที่จะต้องใช้หน่วยการเก็บข้อมูลเป็นจำนวนมาก ฉะนั้นการนำเทคโนโลยีทางด้านวิทยาการคอมพิวเตอร์เข้ามามีส่วนร่วมในการจัดการ และจัดเก็บข้อมูลให้อยู่ในรูปของสื่ออิเลคโทรนิกส์ จึงเป็นอีกทางเลือกหนึ่งที่จะช่วยให้การจัดเก็บข้อมูลมีระบบระเบียบและมีประสิทธิภาพ แต่ถึงแม้ว่าเทคโนโลยีทางด้านอุปกรณ์คอมพิวเตอร์สำหรับการเก็บข้อมูลจะถูกพัฒนาให้มีศักยภาพ และสามารถรองรับปริมาณความต้องการในการเก็บข้อมูลได้ในระดับหนึ่งแล้วก็ตาม แต่แนวโน้มของข้อมูลสารที่เพิ่มขึ้นอยู่ตลอดเวลา อาจส่งผลให้หน่วยการเก็บข้อมูลไม่เพียงพอต่อความต้องการในอนาคต

วิธีการหนึ่งที่เป็นไปได้ คือ การเพิ่มอุปกรณ์ที่ใช้สำหรับเก็บข้อมูลให้กับเครื่องคอมพิวเตอร์ แต่ก็ยังมีปัญหา คือ ต้องเสียค่าใช้จ่ายในการซื้ออุปกรณ์เพิ่มขึ้น ฉะนั้นวิธีการที่เหมาะสมกว่า คือ การลดขนาดของข้อมูลให้มีขนาดเล็กลง เพื่อให้หน่วยการเก็บข้อมูลสามารถเก็บข้อมูลได้มากขึ้น นอกจากการลดขนาดของข้อมูลที่มีผลให้สามารถเก็บข้อมูลได้มากขึ้นแล้วนั้น ผลพลอยได้จากการลดขนาดของข้อมูล คือ เวลาและปริมาณข้อมูลในการรับ-ส่งผ่านช่องสัญญาณระบบการติดต่อสื่อสารก็ลดลงอีกด้วย จากเหตุผลที่กล่าวมา ทำให้เกิดงานวิจัยทางด้านการลดขนาดข้อมูลขึ้นมากมาย โดยส่วนมากจะเป็นงานวิจัยที่เน้นการพัฒนาและปรับปรุงอัลกอริทึมของการบีบอัดข้อมูล(Data Compression) แต่ทว่าปริมาณข้อมูลที่เพิ่มขึ้นอย่างต่อเนื่อง การพัฒนาอัลกอริทึมการบีบอัดข้อมูลเพียงอย่างเดียวจึงไม่เพียงพอต่อความต้องการในการลดปริมาณข้อมูล จึงเกิดแนวคิดที่จะเพิ่มประสิทธิภาพให้กับการบีบอัดข้อมูลในแนวทางอื่น ซึ่งแนวทางที่ว่า คือ การเพิ่มส่วนการแปลง ข้อมูล(Transformation) โดยส่วนการแปลงข้อมูลจะมีหน้าที่จัดการกับข้อมูลในเบื้องต้น เช่น ทำให้ข้อมูลเกิดรูปแบบซ้ำซ้อนเพิ่มมากขึ้น ซึ่งเป็นที่ยอมรับว่าสามารถช่วยให้ประสิทธิภาพการลดขนาดข้อมูลเพิ่มขึ้น

จากที่กล่าวข้างต้น ทำให้เกิดงานวิจัยที่เกี่ยวข้องกับการแปลงข้อมูลขึ้น ซึ่งส่วนหนึ่งของงานวิจัยที่ผ่านมา มุ่งเน้นการแปลงข้อมูลที่เฉพาะเจาะจงกับข้อมูลประเภทข้อความ โดยออกแบบและพัฒนาวิธีการแปลงข้อมูลจากพื้นฐานโครงสร้างข้อมูลภาษาอังกฤษ ซึ่งให้ผลเป็นที่ยอมรับว่าการแปลงข้อมูลที่ใช้ในการลดขนาดข้อมูลประเภทข้อความภาษาอังกฤษมีประสิทธิภาพ แต่อย่างไร

ก็ตาม ในปัจจุบันยังไม่มีงานวิจัยการแปลงข้อมูลที่เกี่ยวข้องโดยตรงกับข้อมูลประเภทข้อความภาษาไทย ผู้วิจัยจึงพัฒนาวิธีการแปลงข้อมูลประเภทข้อความภาษาไทย โดยหวังว่าจะช่วยลดขนาดข้อมูลได้เช่นเดียวกัน

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์นี้มีวัตถุประสงค์เพื่อพัฒนาวิธีการแปลงข้อความภาษาไทย เพื่อใช้ในการเพิ่มประสิทธิภาพในการลดขนาดของข้อมูลประเภทข้อความภาษาไทย

1.3 สมมติฐานของการศึกษา

เมื่อนำข้อมูลประเภทข้อความภาษาไทยมาผ่านการแปลงข้อมูลที่ได้จากวิทยานิพนธ์นี้แล้ว จะทำให้สามารถลดขนาดของข้อมูลได้มากกว่าข้อมูลที่ไม่ได้ผ่านวิธีการแปลงข้อมูล

1.4 ขอบเขตการวิจัย

วิทยานิพนธ์นี้มีขอบเขตการวิจัยดังต่อไปนี้

1. งานวิจัยนี้จะเน้นการแปลงข้อมูลประเภทข้อความภาษาไทยที่มี พยัญชนะ สระ และวรรณยุกต์ ที่ประกอบกันขึ้นเป็นคำ

2. ข้อมูลภาษาไทยที่ใช้ในการทดสอบจะเป็นข้อมูลที่เก็บในรูปของรหัสแอสกี(ASCII Code) และเป็นรหัสภาษาไทยของสำนักงานมาตรฐานผลิตภัณฑ์อุตสาหกรรม(ส.ม.อ.)

3. โปรแกรมต้นแบบที่ได้จากวิทยานิพนธ์นี้ จะเป็นโปรแกรมที่แยกออกจากโปรแกรมการบีบอัดข้อมูล

4. ประสิทธิภาพหรือความเหมาะสมของการแปลงข้อความภาษาไทยที่ได้ จะวัดจากปริมาณของข้อมูลที่ลดลงหลังผ่านการบีบอัดข้อมูลเป็นหลัก

5. ข้อมูลที่ได้จากการแปลงข้อมูลด้วยอัลกอริทึมการแปลงข้อความภาษาไทยจะถูกนำมาทดสอบกับโปรแกรม PKZIP, ARJ, BZIP2 โปรแกรมที่พัฒนาจากอัลกอริทึมฮัฟฟ์แมน และโปรแกรมที่พัฒนาจากอัลกอริทึมวิธีการเชิงคำนวณ

6. ข้อมูลภาษาไทยที่ใช้ในการทดสอบจะได้จากการสุ่มตัวอย่างจากหนังสือ และเอกสารต่างๆ ได้แก่ หนังสือพิมพ์ วารสาร นิตยสาร รายงาน จดหมายราชการ หนังสืออ่านทั่วๆไป ยกเว้นหนังสือประเภทวรรณคดี หรือตำราวิชาการที่แปลมาจากต่างประเทศ จำนวน 100 ตัวอย่าง

1.5 ขั้นตอนของการศึกษา

1. ศึกษาวิธีการแปลงข้อมูลจากงานวิจัยต่างๆ เพื่อหาข้อดี ข้อเสีย และเทคนิคการแปลงข้อมูล

2. ศึกษางานวิจัยที่เกี่ยวข้องกับข้อมูลภาษาไทย

3. ตั้งข้อสมมติฐานการทดลอง

4. ออกแบบโครงสร้างและขั้นตอนของวิธีการแปลงข้อความภาษาไทย โดยพิจารณาจากข้อดี ข้อเสีย ของวิธีการแปลงข้อมูลจากงานวิจัยที่เกี่ยวข้อง

5. ศึกษาภาษาโปรแกรมที่ใช้สำหรับเขียนโปรแกรมต้นแบบ

6. พัฒนาโปรแกรมต้นแบบที่ใช้สำหรับแปลงข้อความภาษาไทย

7. รวบรวมโปรแกรมการบีบอัดข้อมูล ได้แก่ โปรแกรม PKZIP, ARJ, BZIP2 โปรแกรมที่พัฒนาจากอัลกอริทึมฮัฟฟ์แมน และโปรแกรมที่พัฒนาจากอัลกอริทึมวิธีการเชิงคำนวณ พร้อมทั้งรวบรวมข้อมูลภาษาไทยที่ใช้ในการทดสอบ 100 ตัวอย่าง

8. ทำการทดสอบวิธีการแปลงข้อความภาษาไทย ด้วยโปรแกรมการบีบอัดข้อมูลและตัวอย่างข้อมูลที่จัดเตรียมไว้

9. วิเคราะห์ผลการทดลอง เปรียบเทียบปริมาณข้อมูลหลังผ่านการบีบอัดข้อมูลเพียงอย่างเดียว กับปริมาณข้อมูลหลังผ่านทั้งการแปลงข้อความภาษาไทยและการบีบอัดข้อมูล

10. ประเมินผล พร้อมทั้งสรุปผลการทดลอง

11. เขียนวิทยานิพนธ์

บทที่ 2
การแปลงข้อมูลและงานวิจัยที่เกี่ยวข้องกับลักษณะข้อมูลคำไทย

ในการเพิ่มประสิทธิภาพให้กับการลดขนาดข้อมูลสามารถทำได้ 2 แนวทางหลัก คือ ทำการปรับปรุงวิธีการบีบอัดข้อมูลมาตรฐานให้มีประสิทธิภาพมากขึ้น และแปลงข้อมูลให้อยู่ในรูปแบบที่ทำให้อัตราการบีบอัดข้อมูลเพิ่มขึ้น ซึ่งแม้ว่าการปรับปรุงวิธีการบีบอัดข้อมูลมาตรฐานให้มีประสิทธิภาพมากขึ้น จะสามารถเพิ่มประสิทธิภาพให้กับวิธีการลดขนาดของข้อมูลได้ แต่แนวทางนี้ก็เพิ่มความซับซ้อนให้กับตัวอัลกอริทึมใหม่ที่เกิดขึ้นตามไปด้วย และเนื่องจากว่ายังไม่มีอัลกอริทึมใด ที่สามารถลดขนาดของข้อมูลได้โดยไม่ขึ้นกับประเภทและรูปแบบของข้อมูล ดังนั้นเมื่อนำมาประยุกต์ใช้งานจริงจึงเกิดความยุ่งยากในการปรับปรุงและแก้ไข ผู้วิจัยจึงเลือกแนวทางแปลงข้อมูลให้อยู่ในรูปแบบที่ทำให้อัตราการบีบอัดข้อมูลเพิ่มขึ้น หรือที่รู้จักกันดีว่า วิธีการแปลงข้อมูล เพื่อใช้ในเพิ่มประสิทธิภาพการลดขนาดข้อมูลประเภทข้อความภาษาไทย ซึ่งในบทนี้จะกล่าวถึงงานวิจัยที่เกี่ยวข้อง โดยแบ่งเนื้อหาที่เกี่ยวข้องกับงานวิจัยออกเป็น 2 ส่วน ส่วนแรกจะกล่าวถึงการแปลงข้อมูล และส่วนที่สองจะกล่าวถึงลักษณะข้อมูลคำไทย

2.1 วิธีการแปลงข้อมูล

โดยทั่วไปวิธีการแปลงข้อมูล จะมีองค์ประกอบที่สำคัญ 2 ส่วน [10] คือ ส่วนการเข้ารหัสข้อมูล และส่วนการถอดรหัสข้อมูล ซึ่งส่วนการเข้ารหัสข้อมูล จะทำหน้าที่เปลี่ยนข้อมูลให้อยู่ในรูปแบบใดแบบหนึ่ง ที่ทำให้ตัวบีบอัดข้อมูลลดขนาดข้อมูลได้มากขึ้น และส่วนการถอดรหัสข้อมูล จะทำหน้าที่แปลงข้อมูลที่เข้ารหัสแล้วให้กลับมาอยู่ในรูปข้อมูลต้นกำเนิดดังเดิม

2.1.1 วิธีเบอร์โรวส์-วีเลอร์ (Burrows-Wheeler Transformation หรือ BWT)

การแปลงข้อมูลด้วยวิธีเบอร์โรวส์-วีเลอร์ [5, 6] เป็นวิธีที่ทำให้อัตราการบีบอัดข้อมูลเพิ่มขึ้น โดยลักษณะการแปลงข้อมูลวิธีนี้ จะแบ่งข้อมูลต้นกำเนิดออกเป็นกลุ่ม หรือ บล็อก(Block) ที่มีขนาดเท่าๆกัน แล้วทำการเปลี่ยนลำดับของอักษรภายในบล็อกดังกล่าว เพื่อให้อักษรที่เหมือนกันเรียงต่อกัน เช่น bacba ถูกเปลี่ยนเป็น bbcaa เมื่ออักษรที่เหมือนกันเรียงต่อกัน การใช้อัลกอริทึมการบีบอัดข้อมูล เช่น วิธีเข้ารหัสแบบลดความยาว(Run Length Encoding) จะทำให้อัตรา

การลดขนาดข้อมูลเพิ่มขึ้น ซึ่งรายละเอียดการเข้ารหัสและถอดรหัสด้วยวิธีเบอร์โรวส์-วีเลอร์ มีดังต่อไปนี้

2.1.1.1 การเข้ารหัสด้วยวิธีเบอร์โรวส์-วีเลอร์

กำหนดให้ S เป็นบล็อกข้อมูลขนาด N ตัวอักษร โดยที่แต่ละตัวอักษรในบล็อกแทนด้วยสัญลักษณ์ B[0], B[1], B[2], …, B[N-2], B[N-1] ตามลำดับ และกำหนดให้เมตริก M เป็นเมตริกจัตุรัสขนาด N x N ซึ่งมีรูปแบบการเก็บอักษรในเมตริกดังนี้

$$\begin{bmatrix} B[0] & B[1] & B[2] & \dots & B[N-3] & B[N-2] & B[N-1] \\ B[1] & B[2] & B[3] & \dots & B[N-2] & B[N-1] & B[0] \\ B[2] & B[3] & B[4] & \dots & B[N-1] & B[0] & B[1] \\ & \dots & & & \dots & & \\ & \dots & & & \dots & & \\ B[N-2] & B[N-1] & B[0] & \dots & B[N-5] & B[N-4] & B[N-3] \\ B[N-1] & B[0] & B[1] & \dots & B[N-4] & B[N-3] & B[N-2] \end{bmatrix}_{N \times N}$$

แถวแรกของเมตริก M ได้จากการนำเอาอักษรแต่ละตัวในบล็อก S มาวางแต่ละตำแหน่งในแถวตามลำดับ

แถวที่สองของเมตริก M ได้จากการย้ายตำแหน่งตัวอักษรที่อยู่หน้าสุดของแถวแรกไปไว้ในตำแหน่งหลังสุด แล้วเลื่อนอักษรทุกตัวที่เหลือในแถวไปทางซ้าย 1 ตำแหน่ง

แถวที่สามของเมตริก M ได้จากการย้ายอักษรที่อยู่หน้าสุดของแถวที่สองไปไว้ในตำแหน่งหลังสุด แล้วเลื่อนอักษรทุกตัวที่เหลือในแถวไปทางซ้าย 1 ตำแหน่ง โดยจะทำในลักษณะเดียวกันนี้ไปจนครบ N-1 ครั้ง ก็จะได้เมตริก M ขนาด N x N ดังแสดงในตัวอย่างที่ 2.1

<u>ตัวอย่างที่ 2.1</u> สมมติว่าบล็อก S ในข้อมูลต้นกำเนิดมีอักษรภาษาอังกฤษเรียงต่อกัน คือ bacba

เลขแถว	เมตริก M
0	b a c b a
1	a c b a b
2	c b a b a
3	b a b a c
4	a b a c b

จากโจทย์ S = bacba,

N = 5

ข้อมูลแต่ละแถวในเมตริก M จะถูกมองเสมือนว่าเป็นกลุ่มข้อมูลแต่ละกลุ่ม เช่นในตัวอย่างที่ 2.1 แถวทั้ง 5 แถว ในเมตริก M จะถูกมองเสมือนว่าเป็นกลุ่มข้อมูลภาษาอังกฤษ 5 กลุ่ม โดยมีเลขแถวกำกับตั้งแต่ 0 ถึง 4(N=5) คือ bacba, acbab, cbaba, babac และ abacb ตามลำดับ ซึ่งในขั้นตอนต่อไป จะนำเอากลุ่มข้อมูลเหล่านี้ มาเรียงลำดับ(Sorting) ในลักษณะเดียวกับการเรียงคำศัพท์ในพจนานุกรม จากนั้นจะนำกลุ่มข้อมูลทั้งหมดที่เรียงลำดับแล้ว ไปใส่ไว้ในเมตริกใหม่ที่สร้างขึ้น โดยเริ่มใส่ข้อมูลทีละแถวจากแถวบนมายังแถวล่างสุด และเรียกเมตริกใหม่ที่เกิดขึ้นดังกล่าวว่า เมตริก M′ ดังตัวอย่างที่ 2.2

จากนั้นจึงเริ่มใส่กลุ่มข้อมูลทีละกลุ่มลงในเมตริก M′ ทีละแถวจากแถวบนสุดลงมายังแถวล่างสุด ดังตัวอย่างที่ 2.2

ตัวอย่างที่ 2.2 สร้างเมตริก M′ จาก เมตริก M จากตัวอย่างที่ 2.1

เมตริก M มีกลุ่มข้อมูล 5 กลุ่ม และมีลำดับดังนี้ bacba, acbab, cbaba, babac และ abacb

เมื่อเรียงลำดับกลุ่มข้อมูลทั้ง 5 กลุ่มใหม่แล้ว ลำดับใหม่ของกลุ่มข้อมูลที่เกิดขึ้นจะเป็นดังนี้ abacb, acbab, babac, bacba และ cbaba

เลขแถว	เมตริก M
0	b a c b a
1	a c b a b
2	c b a b a
3	b a b a c
4	a b a c b

→

เลขแถว	เมตริก M′
4	a b a c b
1	a c b a b
3	b a b a c
0	b a c b a
2	c b a b a

ให้ L เป็นข้อมูลสดมภ์สุดท้ายของเมตริก M′ ผลลัพธ์ของการเข้ารหัส คือคู่ลำดับ (L, I) โดยที่ I คือ เลขแถวตัวแรกที่พบจากการนำเอาข้อมูลหลักสุดท้ายของแถวแรกในเมตริก M′ ไปค้นหาข้อมูลที่มีค่าตรงกับหลักสุดท้ายของเมตริก M โดยจะเรียก I ว่า ดัชนี ดังตัวอย่างที่ 2.3

ตัวอย่างที่ 2.3 หาผลลัพธ์การเข้ารหัส จากตัวอย่างที่ 2.2

เลขแถว	เมตริก M
0	b a c b a
1	a c b a **b**
2	c b a b a
3	b a b a c
4	a b a c b

→

เลขแถว	เมตริก M′
4	a b a c **b**
1	a c b a **b**
3	b a b a **c**
0	b a c b **a**
2	c b a b **a**

จากตัวอย่างที่ 2.3 ข้อมูลในหลักสุดท้ายของแถวแรกในเมตริก M′ คือ b (‘ b ’ ที่ขีดเส้นใต้ในเมตริก M′) ซึ่งเมื่อนำไปค้นในสดมภ์สุดท้ายของเมตริก M พบว่า ‘ b ’ ตัวแรกที่เจอ(‘ b ’ ตัวหนาในเมตริก M) มีค่าเลขแถวเท่ากับ 1 ฉะนั้น I มีค่าเป็น 1 ผลลัพธ์ของข้อมูลที่ผ่านการเข้ารหัสแล้วจึงเป็น (bbcaa ,1)

2.1.1.2 การถอดรหัสด้วยวิธีเบอร์โรวส์-วีเลอร์

จากการเข้ารหัสในหัวข้อที่ 2.1.1.1 ข้อมูลสดมภ์สุดท้ายของเมตริก M′(คู่ลำดับแรกของผลลัพธ์) จะมีจำนวนตัวอักษรเท่ากับข้อมูลสดมภ์สุดท้ายของเมตริก M′เสมอ เช่น ในตัวอย่างที่ 2.3 จะเห็นว่าสดมภ์แรกและสดมภ์สุดท้ายของเมตริก M′ มีตัวอักษรเท่ากัน คือ ‘ a ’ 2 ตัวเท่ากัน ‘ b ’ 2 ตัวเท่ากัน และมี ‘ c ’ 1 ตัวเท่ากัน แต่จะแตกต่างที่ข้อมูลในสดมภ์แรกของเมตริก M′ จะมีการเรียงลำดับตามตัวอักษร ฉะนั้นเมื่อทราบผลลัพธ์ ซึ่งเป็นสดมภ์สุดท้ายของเมตริก M′ ก็จะสามารถหาค่าของสดมภ์แรกของเมตริก M′ ได้ทันที

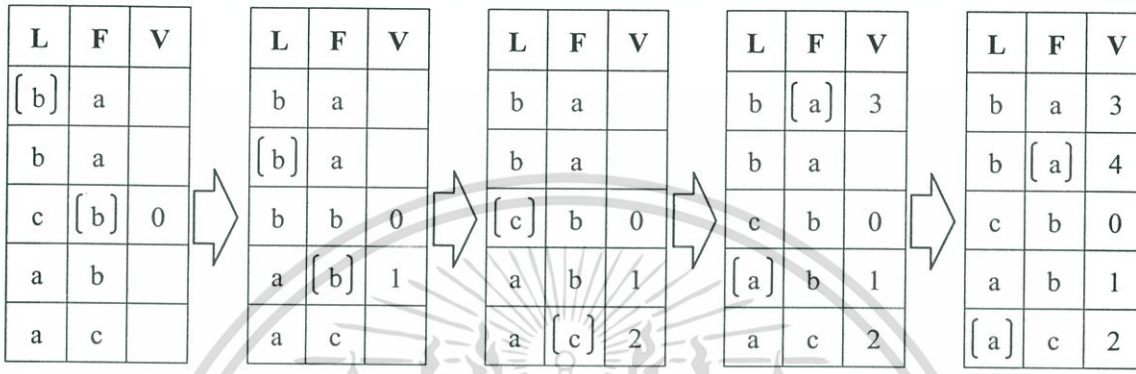
ก่อนการถอดรหัสจะต้องมีข้อมูล 3 ส่วน ได้แก่ สดมภ์สุดท้ายของเมตริก M′(คู่ลำดับแรกของผลลัพธ์) ดัชนี I และเวกเตอร์ V [5] ซึ่งเวกเตอร์ V คือ เมตริกขนาด N x 1 และมีค่าตัวเลขอยู่ในช่วง 0 ถึง N-1 เมื่อ N คือ ขนาดบล็อกข้อมูลของสดมภ์สุดท้ายของเมตริก M′ ซึ่งข้อมูล 2 ส่วนแรก คือ คู่ลำดับของผลลัพธ์ที่ได้จากการเข้ารหัส(ไม่ต้องคำนวณหา) แต่เวกเตอร์ V จะสามารถหาได้จากการคำนวณ โดยมีขั้นตอนดังต่อไปนี้

ในการสร้างเวกเตอร์ V จะนำเอาอักษรทีละตัวในสดมภ์สุดท้ายของเมตริก M′ จากบนลงล่างมาค้นหาอักษรที่มีค่าเดียวกันในสดมภ์แรกของเมตริก M′ เมื่อพบอักษรตัวแรกที่ตรงกัน ก็จะกำหนดค่าตัวเลข ณ แถวเดียวกันกับการพบอักษรที่ตรงกันในสดมภ์แรกของเมตริก M′ โดยให้ตัวแรกที่พบมีค่าเป็น 0 ตัวที่สองมีค่าเป็น 1 ทำเช่นนี้ไปเรื่อยๆ จนกระทั่งถึง N-1 ดังตัวอย่างที่ 2.4

<u>ตัวอย่างที่ 2.4</u> ทำการสร้างเวกเตอร์ V จากคู่ลำดับ (bbcaa ,1)

L = bbcaa, กำหนดให้ F คือ สดมภ์แรกของเมตริก M′

ฉะนั้น F จะมีค่า aabbc เมื่อนำมาทำสร้างเวกเตอร์เพื่อใช้ในการถอดรหัส จะมีลักษณะการทำงานดังต่อไปนี้



ฉะนั้นเวกเตอร์ V = $\begin{bmatrix} 3 \\ 4 \\ 0 \\ 1 \\ 2 \end{bmatrix}$

เมื่อได้เวกเตอร์ในการถอดรหัสแล้ว จะสามารถหาบล็อกข้อมูลต้นกำเนิด(S)ได้ โดยใช้ข้อมูลจากสดมภ์สุดท้ายของเมตริก M′ เวกเตอร์ V และดัชนี I หาค่าอักษรของบล็อกข้อมูลต้นกำเนิดทีละตัว ดังอัลกอริทึมต่อไปนี้

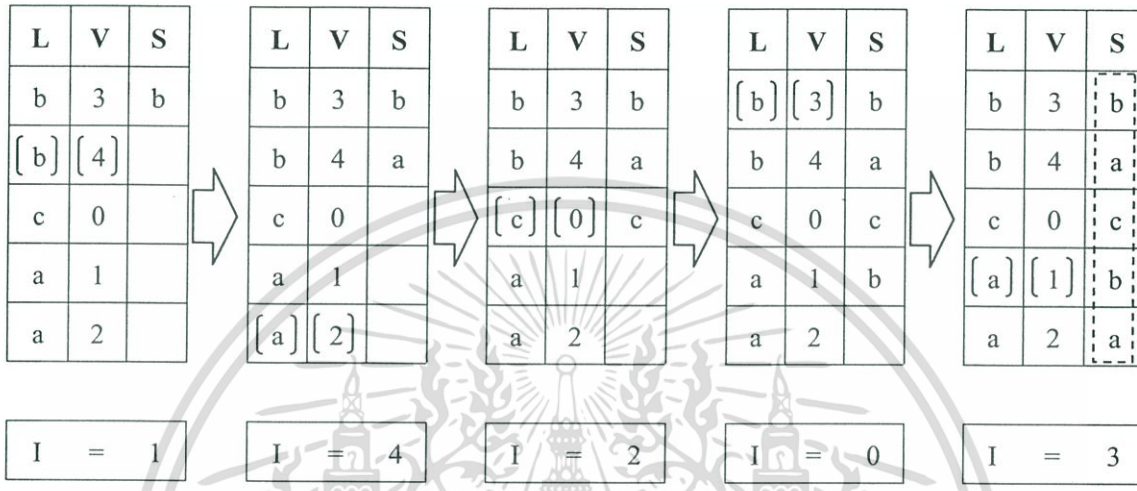
- For j = 0 ถึง N-1
  - เลขแถว = I
  - S[j] = L [เลขแถว]
  - I = V [เลขแถว]

ตัวอย่างที่ 2.5 ทำการหาบล็อกข้อมูลต้นกำเนิด โดยใช้ข้อมูลจากตัวอย่างที่ 2.4

จากคู่ลำดับ (bbcaa ,1)

L = bbcaa, I = 1

การหาบล็อกข้อมูลต้นกำเนิด S จะมีขั้นตอนการทำงานดังต่อไปนี้



บล็อกข้อมูลต้นกำเนิด คือ bacba

จะเห็นได้ว่าการแปลงข้อมูลด้วยวิธีเบอร์โรวส์-วีเลอร์ทำให้ข้อมูลมีขนาดใหญ่ขึ้น เนื่องจากต้องเก็บดัชนีเพื่อใช้ในการถอดรหัสข้อมูล แต่หลังจากนำมาผ่านวิธีการบีบอัดข้อมูล จะทำให้ขนาดของข้อมูลลดลง [5] ไม่เพียงเท่านั้นการแปลงข้อมูลวิธีนี้ยังมีข้อดีที่สามารถใช้กับข้อมูลทุกประเภท แต่อย่างไรก็ตามประสิทธิภาพที่ได้จากการลดขนาดข้อมูลยังคงขึ้นอยู่กับชนิดและลักษณะของข้อมูลที่แตกต่างกันไป ซึ่งในการแปลงข้อมูลด้วยวิธีเบอร์โรวส์-วีเลอร์ จะสามารถลดขนาดข้อมูลได้มากที่สุด เมื่อกำหนดให้ขนาดบล็อกข้อมูลให้ใหญ่กว่าขนาดของข้อมูลต้นกำเนิด [7] ในกรณีที่ข้อมูลต้นกำเนิดเป็นข้อความภาษาอังกฤษ

แม้ว่าการแปลงข้อมูลวิธีนี้จะสามารถลดขนาดข้อมูลได้อย่างมีประสิทธิภาพ แต่เมื่อกำหนดให้ขนาดบล็อกข้อมูลมีขนาดใหญ่ขึ้น เวลาที่ใช้ในการประมวลผลก็จะเพิ่มตามไปด้วย เช่น ถ้าแฟ้มข้อมูลขนาด 12 กิโลไบต์ จะต้องใช้บล็อกข้อมูล(S) ขนาด 12 กิโลไบต์ และประมวลผลด้วยเมตริกขนาด 12 กิโลไบต์ x 12 กิโลไบต์ ซึ่งต้องใช้ทรัพยากรในการประมวลผลอย่างมากในกรณีที่ข้อมูลมีขนาดใหญ่ ฉะนั้นในการแปลงข้อมูลวิธีนี้จึงควรกำหนดขนาดบล็อกให้เหมาะสมกับทรัพยากรที่มีอยู่

เพื่อแก้ปัญหาการในกำหนดขนาดบล็อกข้อมูลของวิธีเบอร์โรวส์-วีเลอร์ จึงเกิดแนวคิดการแปลงข้อมูลที่เฉพาะเจาะจงกับประเภทข้อมูลโดยตรง ซึ่งเป็นอีกทางเลือกหนึ่งที่สามารถช่วยลดขนาดของข้อมูลได้เช่นกัน เช่น ในกรณีที่ข้อมูลมีเนื้อหาเป็นข้อความภาษาอังกฤษ ก็จะพัฒนาการแปลงข้อมูลที่เฉพาะเจาะจงกับข้อความภาษาอังกฤษ โดยจะมีวิธีการต่างๆ ดังจะได้กล่าวต่อไป

2.1.2 วิธีเข้ารหัสดาว (Star Encoding)

วิธีเข้ารหัสดาว [9] เป็นวิธีการแปลงข้อความภาษาอังกฤษ ที่แทนตัวอักษรในคำด้วยสัญลักษณ์ ' * ' โดยที่ D_i เป็นพจนานุกรมที่มีความยาว i ตัวอักษร เมื่อ $i = 1, 2, 3, ..., n$ และกำหนดให้ คำที่ปรากฏในพจนานุกรม D_i ในลำดับที่ j (แทนด้วยสัญลักษณ์ $D_i[j]$) เช่น

$*(D_i[0]) = ***...*$, $*(D_i[1]) = A**...*$,...
$*(D_i[26]) = Z**...*$, $*(D_i[27]) = a**...*$,...
$*(D_i[52]) = z**...*$, $*(D_i[53]) = *A*...*$,...

ถึงแม้การเข้ารหัสวิธีนี้จะแปลงข้อความให้อยู่ในรูป ' * ' ได้มากกว่า 50 เปอร์เซ็นต์ และเพิ่มประสิทธิให้กับการบีบอัดข้อมูลด้วยวิธีฮัฟฟ์แมนและวิธีการเชิงคำนวณ แต่พจนานุกรมที่ใช้ต้องมีขนาดใหญ่ เนื่องจากรหัสที่ใช้แทนคำต้องมีความยาวเท่ากับจำนวนตัวอักษรในคำนั้นๆ ซึ่งทำให้ประสิทธิภาพการแปลงข้อมูลน้อยกว่าวิธีเบอร์โรวส์-วีเลอร์

2.1.3 วิธีดัชนีความยาว (Length Index Preserving Transformation หรือ LIPT)

การแปลงข้อมูลด้วยวิธีดัชนีความยาว [8] เป็นวิธีการแปลงข้อมูลประเภทข้อความภาษาอังกฤษด้วยค่าความยาวคำศัพท์ และตำแหน่งคำศัพท์ภาษาอังกฤษในพจนานุกรม โดยลักษณะการแปลงข้อมูลวิธีการนี้ จะทำการตรวจสอบคำที่ได้จากข้อมูลต้นกำเนิดกับพจนานุกรมคำศัพท์ที่มีขนาดคงที่ หากพบคำใดในพจนานุกรมที่กำหนดไว้ ก็จะทำการแปลงคำดังกล่าวเป็นกลุ่มอักษรพิเศษที่ใช้บอกตำแหน่งและความยาวของคำในพจนานุกรม โดยโครงสร้างการแปลงข้อมูลต้องอาศัยคำศัพท์จากพจนานุกรมเป็นส่วนประกอบหลักสำหรับเข้ารหัสและถอดรหัสข้อมูล ฉะนั้นในคอมพิวเตอร์ทุกเครื่องต้นทางและปลายทางที่จะใช้แปลงข้อมูล จะต้องมีพจนานุกรมที่เหมือนกันเตรียมไว้ล่วงหน้า โดยลักษณะของพจนานุกรมจะถูกแบ่งออกตามความยาวของอักษรในคำ ดังนี้

ให้ D เป็นพจนานุกรมที่มีขนาดคงที่ โดยที่ D_i เป็นพจนานุกรมที่มีความยาว i ตัวอักษร เมื่อ $i = 1, 2, 3, ..., n$ โดยคำที่ปรากฏในพจนานุกรม D_i ในลำดับที่ j (แทนด้วยสัญลักษณ์ $D_i[j]$) จะมีรูปแบบดังต่อไปนี้

$D_i[j]$ = *[ความยาวตัวอักษรในคำ][ตำแหน่งคำในพจนานุกรม]

2.1.3.1 การเข้ารหัสด้วยวิธีดัชนีความยาว

ในการพัฒนาโปรแกรมเพื่อใช้งานจริง ความยาวตัวอักษรในคำ และตำแหน่งคำในพจนานุกรม จะใช้ลำดับของตัวอักษรภาษาอังกฤษแทนค่าตัวเลขตั้งแต่ 1 ถึง 52 (a-z, A-Z) โดยอักษรตัวแรกแทนความยาวของคำที่เก็บในพจนานุกรม และอักษรตัวที่สองแทนตำแหน่งของคำในพจนานุกรม ซึ่งจะแสดงดังตัวอย่างที่ 2.6

ตัวอย่างที่ 2.6 สมมติให้พจนานุกรมบรรจุคำศัพท์ 4 คำ ดังต่อไปนี้ because, biscuit, picture และ someone ซึ่งต้องการที่จะใช้เข้ารหัสข้อความ "I do not like him because of someone"

1) ทำการสร้างพจนานุกรม

คำว่า because, biscuit, picture และ someone มีความยาวคำเท่ากับ 7 ตัวอักษรทั้งหมด ฉะนั้น พจนานุกรมที่ใช้บรรจุคำทั้งหมดจะเป็นพจนานุกรมเดียวกัน(แทนด้วย $D_7[j]$) สำหรับสัญลักษณ์ ที่ใช้แทนคำว่า picture, because, biscuit และ someone ตามลำดับ คือ

$D_7[1]$ = because
$D_7[2]$ = biscuit
$D_7[3]$ = picture
$D_7[4]$ = someone

2) เข้ารหัสข้อความ "I do not like him because of someone"

ในข้อความ "I do not like him because of someone" มีคำที่ตรงกับคำศัพท์ในพจนานุกรมความยาว 7 อักษรอยู่ 2 คำ คือ คำว่า because และคำว่า someone ซึ่งใช้สัญลักษณ์ $D_7[1]$ และ $D_7[4]$

ตามลำดับ ซึ่งหลังจากการเข้ารหัสข้อความ “I do not like him because of someone” รูปแบบข้อมูลที่เข้ารหัสและจะมีรูปแบบดังต่อไปนี้

I do not like him *ga of *gd

จากข้อความด้านบน เลข 7 จะแทนด้วย ‘ g ’ เลข 1 จะแทนด้วย ‘ a ’ และเลข 4 จะแทนด้วย ‘ d ’ ($D_7[1]$ = *ga และ $D_7[4]$ = *gd)

2.1.3.2 การถอดรหัสด้วยวิธีดัชนีความยาว

ในการถอดรหัสการแปลงข้อมูลวิธีนี้ จะทำในทางกลับกัน คือ เมื่ออ่านข้อมูลมาพบสัญลักษณ์ ‘ * ’ ก็จะนำเอาอักษร 2 ตัว ที่อยู่ด้านหลังสัญลักษณ์ ‘ * ’ ไปค้นหาคำในพจนานุกรม เพื่อใช้แปลงข้อมูลกลับ ดังตัวอย่างที่ 2.7

ตัวอย่างที่ 2.7 ถอดรหัสข้อความดังต่อไปนี้ “ I do not like him *ga of *gd ”

นำ ga และ gd ไปตรวจสอบกับพจนานุกรมจะได้ว่า

$D_7[1]$ = because = *ga

$D_7[4]$ = someone = *gd

ฉะนั้นข้อความที่ผ่านการถอดรหัสจะเป็นดังนี้

I do not like him because of someone

การแปลงข้อมูลวิธีนี้ให้อัตราส่วนการลดขนาดข้อมูลมากกว่าวิธีเบอร์โรวส์-วีเลอร์ ในกรณีที่ข้อมูลเป็นข้อความภาษาอังกฤษ เนื่องจากการแปลงข้อมูลวิธีนี้ จะทำให้ข้อมูลมีขนาดเล็กลงเล็กน้อยก่อนนำไปผ่านการบีบอัดข้อมูล อย่างไรก็ตามวิธีการแปลงข้อมูลวิธีนี้ จะต้องใช้พจนานุกรมคำศัพท์ที่มีขนาดแน่นอน และต้องมีคำศัพท์บรรจุในพจนานุกรมเป็นจำนวนมาก เพื่อให้สามารถแปลงข้อมูลคำได้มากตามไปด้วย ฉะนั้นจึงทำให้เกิดข้อจำกัด 3 ประการ [10] ดังต่อไปนี้

1) คำศัพท์บางคำในพจนานุกรมไม่ถูกนำมาใช้ ทำให้สิ้นเปลืองเนื้อที่ในการจัดเก็บ
2) พจนานุกรมต้องมีขนาดใหญ่เกินไป ทำให้ยากต่อการจัดการและแก้ไข
3) ไม่สามารถแปลงคำศัพท์ที่ไม่ได้จัดเก็บไว้ล่วงหน้าได้

2.1.4 วิธีดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัตและพจนานุกรมแบบพลวัต (Semi-Dynamic Length Index Preserving Transformation และ Dynamic Length Index Preserving Transformation)

[10] ได้เสนอวิธีการแปลงข้อมูลที่สามารถลดข้อจำกัดจากการแปลงข้อมูลด้วยวิธีดัชนีความยาวที่กล่าวไว้ในหัวข้อที่ 2.1.3 บางประการ โดยส่วนหนึ่งของงานวิจัย ได้พัฒนาวิธีการแปลงข้อมูลขึ้น 2 วิธี คือ

1) วิธีดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัต
2) วิธีดัชนีความยาวโดยใช้พจนานุกรมแบบพลวัต

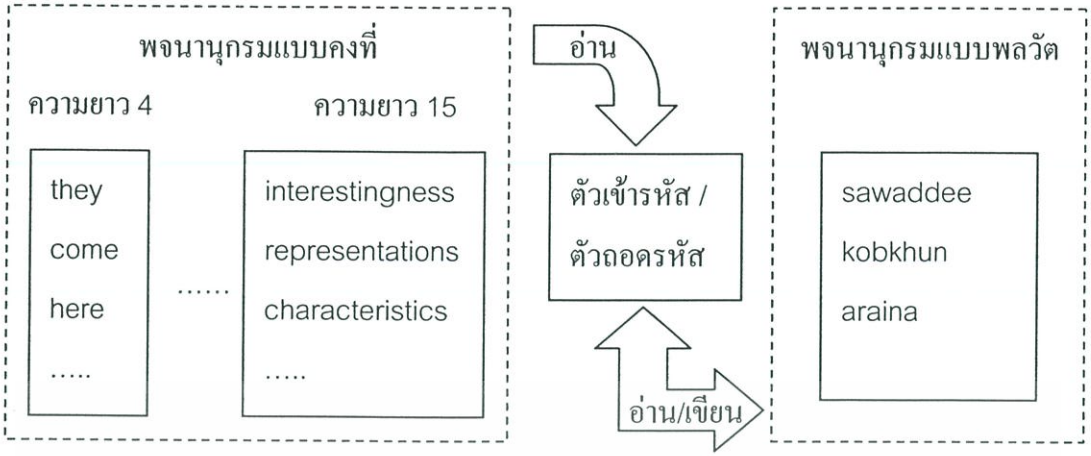
ทั้งวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัต และวิธีดัชนีความยาวพจนานุกรมแบบพลวัตจะใช้พื้นฐานการแปลงข้อมูลในลักษณะเดียวกับวิธีดัชนีความยาว(หัวข้อที่ 2.1.3) แต่จะมีลักษณะของพจนานุกรมที่แตกต่างกัน ดังต่อไป

2.1.4.1 พจนานุกรมของวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัต

วิธีดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัต จะใช้พจนานุกรมเช่นเดียวกับวิธีการแปลงข้อมูลด้วยดัชนีความยาว แต่เพิ่มพจนานุกรมแบบพลวัต(พจนานุกรมที่มีขนาดเปลี่ยนแปลงได้) ขึ้นมาอีก 1 พจนานุกรม ซึ่งพจนานุกรมดังกล่าว จะใช้เก็บคำศัพท์ใหม่จากข้อมูลต้นกำเนิด ดังตัวอย่างที่ 2.8

ตัวอย่างที่ 2.8 สมมติให้พจนานุกรมแบบคงที่มีความยาวคำพจนานุกรมตั้งแต่ 4 ตัวอักษร ถึง 15 ตัวอักษร และในพจนานุกรมดังกล่าวไม่มีคำว่า sawaddee, kobkhun, araina

พจนานุกรมแบบคงที่และแบบพลวัตของวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัต มีลักษณะดังต่อไปนี้



รูปที่ 2.1 พจนานุกรมแบบคงที่ และพจนานุกรมแบบพลวัตของวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัต

จากรูปที่ 2.1 ตัวเข้ารหัสและตัวถอดรหัสสามารถอ่านข้อมูลคำศัพท์จากพจนานุกรมแบบคงที่ได้เพียงอย่างเดียว แต่สามารถอ่านและเขียนข้อมูลคำศัพท์จากพจนานุกรมแบบพลวัตได้ ซึ่งตัวเข้ารหัสแและตัวถอดรหัสจะเขียนคำศัพท์ลงในพจนานุกรมแบบพลวัตขณะทำการแปลงข้อมูล

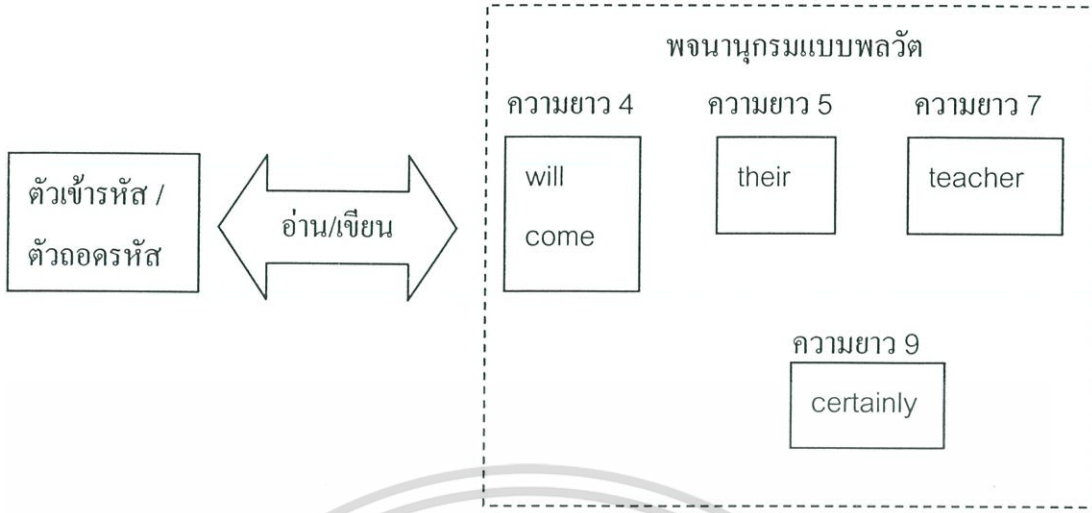
การแปลงข้อมูลวิธีนี้ สามารถแก้ปัญหาการใช้คำศัพท์บางคำซึ่งไม่ได้จัดเก็บไว้ล่วงหน้าของวิธีดัชนีความยาวได้ แต่อย่างไรก็ตาม ยังไม่สามารถแก้ปัญหาคำศัพท์บางคำในพจนานุกรมไม่ถูกนำมาใช้ และพจนานุกรมต้องมีขนาดใหญ่เกินไป

2.1.4.2 พจนานุกรมของวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบพลวัต

วิธีดัชนีความยาวโดยใช้พจนานุกรมแบบพลวัต จะไม่มีพจนานุกรมขนาดคงที่เตรียมไว้ล่วงหน้า แต่จะสร้างพจนานุกรมแบบพลวัตขึ้นในขณะที่ทำการแปลงข้อมูล ซึ่งขั้นตอนการสร้างพจนานุกรมจะทำในลักษณะเดียวกับการแปลงข้อมูลด้วยวิธีดัชนีความยาว แต่คำศัพท์ที่ได้ทั้งหมดจะได้จากข้อมูลต้นกำเนิดเท่านั้น ดังตัวอย่างที่ 2.9

ตัวอย่างที่ 2.9 สมมติต้องการเข้ารหัสข้อความ “Their teacher will certainly come here”

พจนานุกรมแบบพลวัตของวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบพลวัต จะมีลักษณะดังต่อไปนี้



รูปที่ 2.2 พจนานุกรมแบบพลวัตของวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบพลวัต

การแปลงข้อมูลวิธีนี้ สามารถลดข้อจำกัดทั้ง 3 ประการที่กล่าวไปแล้วในหัวข้อที่ 2.1.3 แต่ก็ทำให้ผลเสียบางประการ โดยจะกล่าวในตอนท้ายของหัวข้อนี้

2.1.4.3 การเข้ารหัสด้วยวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัตและพจนานุกรมแบบพลวัต

ขั้นตอนการเข้ารหัสการแปลงข้อมูลประเภทข้อความภาษาอังกฤษด้วยดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัตและพจนานุกรมแบบพลวัตทั้ง 2 วิธี ซึ่งมีลักษณะคล้ายกับวิธีดัชนีความยาว แต่จะแบ่งขั้นตอนการทำงานออกเป็น 3 ขั้นตอนหลัก ดังต่อไปนี้

1) การวิเคราะห์คำ(Analytical Word)
2) การค้นหาดัชนี(Index Searching)
3) การแปลงข้อมูล(Transforming)

การวิเคราะห์คำ จะทำหน้าที่ค้นหาคำจากข้อมูลต้นกำเนิดที่มีความยาวตัวอักษรมากกว่า 4 ตัวอักษรเพื่อนำมาทำการเข้ารหัส โดยสาเหตุที่เข้ารหัสคำที่มีความยาวตั้งแต่ 4 ตัวอักษรขึ้นไป เนื่องจากการเข้ารหัสด้วยวิธีการนี้ จะเปลี่ยนคำในข้อมูลต้นกำเนิดให้มีความยาว 3 ตัวอักษร ฉะนั้นเพื่อให้สามารถลดขนาดของข้อมูลด้วยส่วนหนึ่ง คำที่จะนำมาเข้ารหัสจึงต้องมีขนาดตั้งแต่ 4 ตัวอักษรขึ้นไป ดังตัวอย่างที่ 2.10

การค้นหาดัชนี จะนำคำที่ได้จากส่วนการวิเคราะห์คำมาตรวจสอบกับคำในพจนานุกรมที่เตรียมไว้ล่วงหน้า ดังตัวอย่างที่ 2.11

การแปลงข้อมูล จะทำหน้าที่เข้ารหัสคำในข้อมูลต้นกำเนิด ดังตัวอย่างที่ 2.12

ตัวอย่างที่ 2.10 สมมติว่าข้อมูลต้นกำเนิด คือ

I **really love this organization**

คำที่มีความยาวมากกว่า 4 ตัวอักษร(คำที่ขีดเส้นใต้) จะเป็นคำที่จะต้องถูกนำมาเข้ารหัส

ตัวอย่างที่ 2.11 สมมติว่าต้องการค้นหาดัชนีของคำว่า *organization*

เนื่องจากคำว่า *organization* มีความยาว 12 ตัวอักษร ดังนั้นจึงต้องหาคำดังกล่าวในพจนานุกรมที่มีความยาวคำ 12 ตัวอักษร

j	$D_{12}[j]$
1	casterbridge
2	unsuccessful
3	occasionally
4	applications
5	conversation
6	*organization* → $D_{12}[6]$

จากตัวอย่าง พบคำว่า *organization* ในตำแหน่งที่ 6 ของพจนานุกรม(สัญลักษณ์ที่ใช้แทนคำที่อยู่ในลำดับที่ 6 ของพจนานุกรมที่มีความยาวคำ 12 ตัวอักษร คือ $D_{12}[6]$)

ตัวอย่างที่ 2.12 จากตัวอย่างที่ 2.11 จะเข้ารหัสคำว่า *organization* ในข้อมูลต้นกำเนิด โดยใช้สัญลักษณ์ ' * ' เป็นตัวบอกการเริ่มการแปลงข้อมูล และ ใช้ลำดับของตัวอักษรภาษาอังกฤษแทนค่าตัวเลขตั้งแต่ 1 ถึง 52(a-z, A-Z) โดยอักษรตัวแรกแทนความยาวของคำที่เก็บในพจนานุกรม และ ตัวที่ 2 แทนตำแหน่งของคำในพจนานุกรม ดังนี้

$D_{12}[6]$ = ***lf**

จากตัวอย่างที่ 2.12 เลข 12 จะแทนด้วย ' l ' และเลข 6 จะแทนด้วย ' f '

2.1.4.4 การถอดรหัสด้วยวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัตและพจนานุกรมแบบพลวัต

สำหรับการถอดรหัสจะใช้เทคนิคเดียวกับวิธีดัชนีความยาวในหัวข้อที่ 2.1.3

2.1.4.5 สรุปผลวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัตและพจนานุกรมแบบพลวัต

จากผลการทดลองใน [10] พบว่าการแปลงข้อมูลด้วยวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัต และแบบพลวัต สามารถลดขนาดของข้อมูลได้ โดยการแปลงข้อมูลด้วยวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัต สามารถลดขนาดข้อมูลได้มากกว่าการแปลงข้อมูลด้วยวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบพลวัต ซึ่งข้อดีและข้อด้อยของทั้ง 2 วิธีสามารถสรุปได้ดังนี้

ทั้งวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัตและแบบพลวัต ต้องส่งพจนานุกรมสร้างขึ้นใหม่ไปกับแฟ้มข้อมูลที่แปลงแล้ว เพื่อให้สามารถถอดรหัสข้อมูลได้ ทำให้ต้องใช้เวลาในการถอดรหัสเพิ่มมากขึ้น

การแปลงข้อมูลด้วยวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบกึ่งพลวัต จะใช้พื้นฐานการสร้างพจนานุกรมจากวิธีดัชนีความยาว ซึ่งไม่สามารถแก้ปัญหาขนาดที่ใหญ่เกินไปของพจนานุกรมได้ และหากข้อมูลต้นกำเนิดมีขนาดเล็ก หรือมีการใช้คำในพจนานุกรมในพจนานุกรมแบบพลวัตไม่บ่อยครั้ง จะทำให้ประสิทธิภาพการลดขนาดข้อมูลเพิ่มขึ้นไม่มากนัก

การแปลงข้อมูลด้วยวิธีดัชนีความยาวโดยใช้พจนานุกรมแบบพลวัตสามารถลดขนาดข้อมูลสามารถแก้ไขปัญหาได้ทั้ง 3 ประการ แต่ประสิทธิภาพการลดขนาดข้อมูลเพิ่มขึ้นเพียงเล็กน้อยเท่านั้น หรือในบางกรณีที่ข้อมูลมีขนาดเล็ก ก็จะทำให้ประสิทธิภาพการลดขนาดข้อมูลลดลง

ฉะนั้นการกำหนดพจนานุกรมแบบคงที่ให้มีขนาดเล็ก แต่สามารถใช้พจนานุกรมดังกล่าวให้เกิดประโยชน์สูงสุด ซึ่งการใช้คำจากที่เกิดบ่อยจากสถิตินำจะเป็นแนวทางในการลดปัญหาทั้ง 3 ประการดังกล่าว แต่ยังคงไว้ซึ่งประสิทธิภาพเชิงปริมาณ และประสิทธิภาพเชิงเวลา สำหรับการลดขนาดข้อมูล

2.1.4.6 การนำวิธีการแปลงข้อมูลมาใช้กับข้อความภาษาไทย

วิธีการแปลงข้อมูลที่กล่าวมาในตอนต้น เป็นวิธีการแปลงข้อมูลประเภทข้อความที่ใช้สำหรับข้อมูลประเภทข้อความภาษาอังกฤษ ซึ่งข้อความในภาษาไทยมีลักษณะโครงสร้างคำ และประโยคแตกต่างจากข้อความในภาษาอังกฤษ เช่น

ภาษาอังกฤษมีการแบ่งคำโดยใช้เว้นวรรคแต่ภาษาไทยจะเขียนคำติดกัน

ภาษาอังกฤษมีสระเพียง 5 ตัวและไม่มีวรรณยุกต์ แต่ภาษาไทยมีวรรณยุกต์และสระรวมกันมากกว่า 20 ตัว

ภาษาอังกฤษไม่มีการแยกสระออกจากตัวพยัญชนะเช่นเดียวกับภาษาไทย

อักษร 1 ตัวในภาษาอังกฤษมีตัวเขียนเล็กและตัวเขียนตัวใหญ่ แต่อักษร 1 ตัวของภาษาไทยมีรูปแบบเดียว

ฉะนั้นในการพัฒนาวิธีการแปลงข้อความภาษาไทย จะต้องคำนึงถึงกฎเกณฑ์และลักษณะธรรมชาติทางภาษา รวมถึงลักษณะข้อมูลคำไทยอีกด้วย

2.2 ลักษณะข้อมูลคำไทย

ในส่วนนี้จะกล่าวถึงงานวิจัยการที่เกี่ยวข้องกับการนำภาษาไทยไปใช้กับคอมพิวเตอร์ โดยจะแบ่งออกเป็น 2 หัวข้อ คือ การวิเคราะห์ข้อมูลคำไทย และการเก็บข้อมูลคำไทย

2.2.1 การวิเคราะห์ข้อมูลคำไทย

งานวิจัยที่เกี่ยวข้องกับการวิเคราะห์ข้อมูลคำไทย [1, 2] เป็นงานวิจัยที่ทำการหาสถิติข้อมูลคำไทยที่ใช้ในชีวิตประจำวัน โดยส่วนหนึ่งของงานวิจัยจะแสดงอยู่ในรูปของตารางคำไทยที่จัดเรียงลำดับตามความถี่ของการใช้งานจากมากไปหาน้อย ที่รวบรวมโดยการสุ่มตัวอย่างจากหนังสือและเอกสารต่างๆ ได้แก่ หนังสือพิมพ์ วารสาร นิตยสาร รายงาน จดหมายราชการ หนังสืออ่านทั่วๆไป ยกเว้นหนังสือประเภทวรรณคดี หรือตำราวิชาการที่แปลมาจากต่างประเทศ ดังแสดงไว้ในตารางที่ ก1 ในภาคผนวก ก ซึ่งคำไทยจากสถิติทั้งหมดมีจำนวนคำไม่มากนัก และจำนวนตัวอักษรในคำมีความยาวตั้งแต่ 1 ถึง 8 ตัวอักษร โดยมีการเกิดคำดังกล่าวในข้อความภาษาไทยทั่วๆไปมากกว่าร้อยละ 70 ของข้อมูลทั้งหมด ทำให้สามารถคาดคะเนได้ว่า หากนำคำที่ได้จากสถิติข้อมูลคำไทยนี้ ไปผ่านการเข้ารหัสการแปลงข้อความภาษาไทย น่าจะทำให้การบีบอัดข้อมูลมีประสิทธิภาพมากขึ้น ซึ่งในวิทยานิพนธ์เล่มนี้จะใช้คำจากสถิติข้อมูลคำไทยในการแปลงข้อมูลประเภทข้อความภาษาไทย โดยจะกล่าวในบทต่อไป

2.2.2 การเก็บข้อมูลคำไทย

จากงานวิจัยโครงสร้างข้อมูลที่ใช้เก็บคำในภาษาไทย [3, 4] พบว่าโครงสร้างข้อมูลที่มีลักษณะเป็นสายข้อมูล(Data stream) แบบหนึ่งที่เรียกว่า โครงสร้างข้อมูลดัชนีตาราง(Table Index) เป็นโครงสร้างข้อมูลที่สะดวกและง่ายในการนำไปประยุกต์ใช้งาน อีกทั้งยังใช้เวลาในการสร้างพจนานุกรมไม่มากนัก โดยลักษณะการบรรจุคำไทยแต่ละคำลงในโครงสร้างข้อมูลดัชนีตาราง จะต้องทำการใส่ข้อมูลลงไปในสายข้อมูล 2 ส่วน ส่วนแรกเป็นตัวเลขที่ใช้บอกจำนวนตัวอักษรของคำ อีกทั้งใช้เป็นตัวคั่นคำไทยแต่ละคำในสายข้อมูล คือ เมื่อพบตัวเลขก็จะทราบทันทีว่าอักษรที่ต่ออยู่ด้านหลังเป็นกลุ่มของอักษรของคำไทย ส่วนที่สอง คือ คำไทยที่มีความยาวเท่ากับตัวเลขที่ใช้บอกจำนวนตัวอักษรนั้น ซึ่งคำที่จะนำมาบรรจุในพจนานุกรมประเภทนี้ จะเรียงลำดับคำตามหลักการเรียงอักษรไทย เพื่อให้ง่ายต่อการค้นหาในกรณีที่มีคำเป็นจำนวนมาก ดังแสดงในตัวอย่างที่ 2.13

ตัวอย่างที่ 2.13 สมมติว่าจะเก็บคำไทยลงในพจนานุกรมที่มีโครงสร้างข้อมูลดัชนีตาราง จำนวน 6 คำ คือ คำว่า *การ จะ ได้ ที่ ใน* และ *เป็น* คำไทยทั้งหมดจะถูกเก็บในลักษณะของสายข้อมูล เช่นนี้

3การ2จะ3ได้3ที่2ใน4เป็น

จากตัวอย่างที่ 2.13 แสดงการเก็บคำทั้ง 6 คำ คือคำว่า *การ จะ ได้ ที่ ใน* และ *เป็น* ลงในพจนานุกรมที่ใช้โครงสร้างข้อมูลแบบดัชนีตาราง โดยตัวเลข ‘ 3 ’ ที่วางไว้หน้าคำว่า “การ” คือ ตัวเลขที่บอกจำนวนตัวอักษรของคำว่า “การ” ตัวเลข ‘ 2 ’ คือ ตัวเลขที่บอกจำนวนตัวอักษร และเป็นคั่นระหว่างคำว่า “การ” และคำว่า “จะ” ซึ่งคำทั้ง 6 คำจะถูกเก็บลงในพจนานุกรม ตามลำดับ

ผู้วิจัยได้นำความรู้จากการศึกษางานวิจัยการแปลงข้อมูลประเภทข้อความภาษาอังกฤษ และลักษณะของข้อมูลคำไทย มาประยุกต์ใช้ในการพัฒนาวิธีการแปลงข้อความภาษาไทย ซึ่งรายละเอียดต่างๆจะกล่าวในบทต่อไป

บทที่ 3
วิธีการแปลงข้อความภาษาไทย

จากการศึกษางานวิจัยในบทที่ 2 ทำให้ทราบถึง ข้อดี ข้อเสีย และเทคนิคต่างๆ ของวิธีการแปลงข้อมูลประเภทข้อความภาษาอังกฤษ ซึ่งเป็นประโยชน์ในการออกแบบและพัฒนาวิธีการแปลงประเภทข้อความภาษาไทย โดยแนวทางในการพัฒนาวิธีการแปลงข้อความภาษาไทย จะมีวัตถุประสงค์หลักในการพัฒนา 2 ประการ คือ ต้องการทำให้ข้อมูลเกิดความซ้ำซ้อนเพิ่มมากขึ้น และต้องการทำให้ตัวเข้ารหัสการแปลงข้อความมีส่วนช่วยลดปริมาณข้อมูลในเบื้องต้น ซึ่งผู้วิจัยได้พัฒนาเป็นโปรแกรมต้นแบบสำหรับวิธีการแปลงข้อความภาษาไทยเพื่อใช้ในการทดลอง โดยเริ่มแรกผู้วิจัยได้พัฒนาวิธีการแปลงข้อความภาษาไทยโดยนำเอาคำที่มีการบันทึกไว้ในสถิติทั้งหมดเก็บลงในพจนานุกรมที่ใช้สำหรับเข้ารหัสและถอดรหัส ซึ่งพบว่าวิธีนี้มีความยืดหยุ่นในการเพิ่มและลดคำศัพท์ในพจนานุกรมได้ แต่มีส่วนช่วยให้ปริมาณข้อมูลลดลงเพียงเล็กน้อยเท่านั้น ดังนั้นผู้วิจัยจึงพัฒนาวิธีการแปลงข้อความภาษาไทยขึ้นอีก 2 วิธี โดยทั้ง 2 วิธีจะใช้คำจากสถิติเพียงบางส่วนเก็บลงในพจนานุกรม และยังคงใช้หลักการในการเข้ารหัสและถอดรหัสเช่นเดิม แต่จะเน้นให้สามารถลดขนาดข้อมูลในเบื้องต้นเพิ่มมากขึ้น อย่างไรก็ตาม การแปลงข้อความทั้ง 2 วิธีนี้ ทำให้เกิดข้อจำกัดขึ้นหลายประการ ซึ่งรายละเอียดต่างๆ จะกล่าวเป็นลำดับต่อไป โดยแบ่งออกเป็น 2 ส่วน คือ ส่วนการสร้างพจนานุกรมคำไทย และส่วนการแปลงข้อความภาษาไทย

3.1 การสร้างพจนานุกรมคำไทย

ก่อนทำการแปลงข้อความภาษาไทย คอมพิวเตอร์ที่ใช้ในการเข้ารหัสและถอดรหัสการแปลงข้อความภาษาไทย จะต้องมีพจนานุกรมคำไทยที่ใช้สำหรับอ้างอิงการเข้ารหัสและถอดรหัสไว้ล่วงหน้า ซึ่งคำที่จะใช้บรรจุในพจนานุกรมคำไทยของการแปลงข้อความภาษาไทยนี้ จะใช้คำที่ได้จากงานวิจัยการวิเคราะห์ข้อมูลคำไทยทั้งหมด [1, 2] จำนวน 511 คำ ดังแสดงไว้ในตารางที่ ก1 ในภาคผนวก ก เนื่องจากมีจำนวนคำตามสถิติไม่มากนัก จึงไม่มีความจำเป็นที่ต้องเก็บคำดังกล่าวลงในโครงสร้างข้อมูลที่มีความซับซ้อนและยากต่อการจัดการ แม้ว่าโครงสร้างข้อมูลดังกล่าวจะใช้พื้นที่ในการบรรจุคำน้อยกว่าก็ตาม เช่น โครงสร้างต้นไม้(Tree structure) ฉะนั้นโครงสร้างของพจนานุกรมคำไทย ที่ใช้ในการอ้างอิงการเข้ารหัสและถอดรหัสการแปลงข้อความภาษาไทยในวิทยานิพนธ์เล่มนี้ จึงเลือกใช้โครงสร้างข้อมูลที่มีลักษณะง่ายและสะดวกต่อการจัดเก็บ นั่นก็คือโครงสร้างข้อมูลแบบสายข้อมูล

จากงานวิจัยโครงสร้างข้อมูลที่ใช้เก็บคำในภาษาไทย [4] ที่กล่าวไปแล้วในบทที่ 2 พบว่าโครงสร้างข้อมูลที่มีลักษณะเป็นสายข้อมูลแบบหนึ่งที่เรียกว่า โครงสร้างข้อมูลดัชนีตารางเป็นโครงสร้างข้อมูลที่สะดวกและง่ายในการนำไปประยุกต์ใช้งาน อีกทั้งยังใช้เวลาในการสร้างพจนานุกรมน้อยอีกด้วย อย่างไรก็ตามเพื่อให้เกิดความสะดวก และง่ายต่อการนำมาประยุกต์ใช้กับการแปลงข้อ-ความภาษาไทย(หัวข้อที่ 3.2) ผู้ทำวิทยานิพนธ์จึงกำหนดลักษณะการเก็บข้อมูลในพจนานุกรมคำไทยให้แตกต่างเล็กน้อย คือ เปลี่ยนตัวเลขที่ใช้บอกความยาวคำในโครงสร้างข้อมูลดัชนีตารางเป็น รหัสตัวหนึ่งที่เรียกว่า รหัสแทนคำ ซึ่งรหัสแทนคำนี้จะเป็นค่าที่มีความเป็นหนึ่งเดียว(unique) กล่าวอีกนัยหนึ่งคือ แต่ละรหัสแทนคำจะเกิดขึ้นในพจนานุกรมคำไทยได้เพียงครั้งเดียวเท่านั้น ฉะนั้นในการบรรจุคำไทยแต่ละคำลงในพจนานุกรมประเภทนี้ จะมีข้อมูลที่ต้องใส่ลงไป 2 ส่วน คือ ส่วนที่เป็นรหัสแทนคำ และส่วนที่เป็นคำไทย ดังรูปแบบต่อไปนี้

[รหัสแทนคำ][คำไทย]

รหัสแทนคำแต่ละตัวจะมีคุณสมบัติในการอ้างอิงคำไทยแต่ละคำในพจนานุกรม อีกทั้งทำหน้าที่เป็นตัวคั่นคำที่เรียงต่อกันในสายข้อมูล

สำหรับการกำหนดขนาดของรหัสแทนคำ และคำไทย จะขึ้นอยู่กับจำนวนคำที่จะบรรจุในพจนานุกรม กล่าวคือ รหัสแทนคำจะมีขนาดเล็กเมื่อกำหนดให้คำไทยในพจนานุกรมมีปริมาณน้อย ในทางกลับกัน รหัสแทนคำจะมีขนาดใหญ่เมื่อคำไทยในพจนานุกรมมีปริมาณมาก ซึ่งจากสถิติข้อมูลคำไทยพบว่า คำไทยที่มีการบันทึกไว้ มีจำนวนทั้งสิ้น 511 คำ ฉะนั้นการกำหนดรหัสแทนคำให้คำไทยแต่ละคำต้องใช้หน่วยความจำอย่างน้อย 2 ไบต์ จึงจะสามารถรองรับคำไทยได้ทั้งหมด อย่างไรก็ตาม แม้ว่าการใช้รหัสแทนคำขนาดไบต์เดียว จะสามารถรองรับคำไทยได้เพียงบางส่วน แต่ในการนำมาประยุกต์ใช้กับวิธีการแปลงข้อความภาษาไทย(รายละเอียดในหัวข้อที่ 3.2.2) จะช่วยลดขนาดข้อมูลในขณะที่ทำการแปลงมูลได้เป็นอย่างมาก ผู้ทำวิทยานิพนธ์จึงกำหนดให้จำนวนคำศัพท์ใช้บรรจุไว้ในพจนานุกรมมีจำนวน 109 คำ(รายละเอียดในหัวข้อที่ 3.1.1) แล้วกำหนดทิศทางการออกแบบและทดลองการแปลงข้อความภาษาไทยออกเป็น 2 แนวทางหลัก คือ การแปลงข้อความภาษาไทยด้วยพจนานุกรมที่ใช้รหัสแทนคำขนาด 1 ไบต์ และการแปลงข้อความภาษาไทยด้วยพจนานุกรมที่ใช้รหัสแทนคำขนาด 2 ไบต์ โดยจะเรียกการแปลงข้อความภาษาไทยทั้ง 2 วิธี ว่า การแปลงข้อความด้วยพจนานุกรม 1 ไบต์ และการแปลงข้อความด้วยพจนานุกรม 2 ไบต์ ตามลำดับ และเรียกพจนานุกรมที่ใช้รหัสแทนคำขนาด 1 ไบต์ และ 2 ไบต์ ว่า พจนานุกรม 1 ไบต์ และพจนานุกรม 2 ไบต์ ตามลำดับ เช่นกัน

หลักการคร่าวๆของพจนานุกรมคำไทย 1 ไบต์ และ 2 ไบต์ มีดังต่อไปนี้

ลักษณะการเรียงลำดับคำที่จะบรรจุในพจนานุกรมคำไทยทั้ง 2 แบบนี้ จะเรียงลำดับตามความยาวของคำจากมากไปหาน้อยเป็นลำดับแรก แล้วจึงเรียงตามความน่าจะเป็นของคำที่จะเกิดขึ้นในกรณีที่ความยาวของคำมีขนาดเท่ากันเป็นลำดับต่อไป โดยไม่เรียงลำดับคำตามหลักการเรียงอักษรภาษาไทยเช่นเดียวกับโครงสร้างดัชนีตาราง สาเหตุที่ต้องเปลี่ยนการเรียงลำดับคำให้อยู่ในลักษณะนี้ จะกล่าวถึงในหัวข้อที่ 3.2 การแปลงข้อความภาษาไทย ต่อไป

รหัสแทนคำ 1 รหัส จะใช้เป็นตัวแทนของคำ 1 คำ และรหัสแทนคำที่ใช้ จะมีความยาวคงที่ตลอดทั้งพจนานุกรม(Fixed Length) คือ รหัสแทนคำจะใช้หน่วยความจำ 1 ไบต์ ในกรณีพจนานุกรม 1 ไบต์ รหัสแทนคำใช้หน่วยความจำ 2 ไบต์ ในกรณีพจนานุกรม 2 ไบต์

การเก็บคำไทย 1 คำ ลงในพจนานุกรม 1 ไบต์ จะใช้หน่วยความจำในการเก็บอย่างน้อยที่สุด 2 ไบต์ โดยไบต์แรก คือ รหัสแทนคำ และไบต์ต่อๆไป คือ คำที่มีความยาวตั้งแต่ 1 ไบต์ขึ้นไป

การเก็บคำไทย 1 คำ ลงในพจนานุกรม 2 ไบต์ จะใช้หน่วยความจำในการเก็บอย่างน้อยที่สุด 3 ไบต์ โดย 2 ไบต์แรก คือ รหัสแทนคำ และไบต์ต่อๆไป คือ คำที่มีความยาวตั้งแต่ 1 ไบต์ขึ้นไป

ในกรณีที่ใช้คำจากสถิติข้อมูลคำไทยไม่ครบทุกคำ การเลือกคำที่ใช้บรรจุลงในพจนานุกรมจะเลือกตามค่าความถี่ของสถิติข้อมูลคำไทยจากมากไปหาน้อย

3.1.1 พจนานุกรม 1 ไบต์

เนื่องจากรหัสแทนคำแต่ละตัวจะมีคุณสมบัติในการอ้างอิงคำไทยแต่ละคำในพจนานุกรมอีกทั้งทำหน้าที่เป็นตัวค้นคำที่เรียงต่อกันในสายข้อมูล ฉะนั้นเพื่อให้สามารถแยกได้ว่ารหัสแอสกีดังกล่าวเป็นรหัสแทนคำหรือรหัสแอสกีของอักษรไทย รหัสแทนคำทุกตัวในพจนานุกรมต้องไม่มีค่าที่ตรงกับรหัสแอสกีที่อยู่ในกลุ่มของอักษรไทย

จากข้อมูลข้างต้น ทำให้ทราบว่าการสร้างพจนานุกรมคำไทย 1 ไบต์ ไม่สามารถนำเอารหัสแอสกีขนาด 1 ไบต์ ทั้ง 256 ค่า (2^8 ค่า) มาใช้เป็นรหัสแทนคำ ฉะนั้นผู้ทำวิทยานิพนธ์จึงกำหนดให้รหัสแอสกีที่จะสามารถนำมาใช้เป็นรหัสแทนคำ มีจำนวน 109 รหัส กล่าวอีกนัยหนึ่งคือ จำนวนคำที่สามารถบรรจุลงในพจนานุกรมลักษณะนี้ มีได้ไม่เกิน 109 คำ โดยรหัสแอสกีที่จะนำมาใช้เป็นรหัสแทนคำทั้ง 109 ค่า จะเป็นรหัสแอสกีที่อยู่ในกลุ่มของอักษรและสัญลักษณ์ที่ใช้ในภาษาอังกฤษ (เนื่องจากไม่มีอักษรภาษาอังกฤษในคำไทย) และรหัสแอสกีที่ใช้กับระบบการติดต่อสื่อสาร(แม้ว่าข้อมูลที่เข้ารหัสจะมีรหัสแอสกีในกลุ่มของระบบการติดต่อสื่อสาร แต่รูปแบบการติดต่อสื่อสารพื้นฐาน เช่น TCP/IP จะไม่ตีความเนื้อหาสาระของตัวข้อมูล ทำให้ข้อมูลมีความถูกต้องเช่นเดิมเสมอ) ซึ่งมีค่าในรูปของเลขฐาน 16 ตั้งแต่ 10 ถึง 7D(เลขฐาน 16) ยกเว้น 20(เว้นวรรค) โดยที่รหัส

แอสกีที่เหลืออีก 147 รหัส จะไม่ถูกนำมาใช้ ซึ่งรหัสดังกล่าวเป็นรหัสที่อยู่ในกลุ่มของอักษรภาษาไทย กลุ่มของอักษรพิเศษ ตัวอักษรเว้นวรรค และ 7E(เลขฐาน 16) สำหรับเหตุผลที่ไม่ใช้ตัวเว้นวรรคเป็นรหัสแทนคำเช่นกัน เนื่องจากว่าตามปกติแล้ว ตัวเว้นวรรคถือว่าเป็นตัวอักษรในภาษาไทยเช่นกัน คือ เป็นอักษรที่มีหน้าที่คั่นประโยคในข้อความภาษาไทย

สำหรับกรณีของรหัสแอสกี 7E(เลขฐาน 16) จะไม่ถูกนำมากำหนดเป็นรหัสแทนคำ แม้ว่ารหัสแทนรหัส 7E จะอยู่ในกลุ่มของอักษรภาษาอังกฤษ เนื่องจากรหัส 7E จะถูกใช้เป็นรหัสที่ช่วยในการแปลงข้อความภาษาไทย(รายละเอียดในหัวข้อที่ 3.2.2)

การกำหนดรหัสแทนคำให้กับคำไทยแต่ละคำ จะกำหนดรหัสแทนคำที่เรียงลำดับจากน้อยไปหามาก โดยเริ่มจากรหัสที่มีค่าในรูปของเลขฐาน 16 ตั้งแต่ 10, 11, 12, ... ถึง 7D ยกเว้น รหัสที่มีค่า 20(ตัวเว้นวรรค) แล้วนำเอาคำไทยทุกคำที่กำหนดค่ารหัสแทนคำให้เรียบร้อยแล้วมาเรียงต่อกันเป็นสายข้อมูล ดังแสดงในตัวอย่างที่ 3.1

ตัวอย่างที่ 3.1 สมมติว่าจะเก็บคำไทยลงในพจนานุกรมคำไทยที่ใช้รหัสแทนคำ 1 ไบต์ จำนวน 6 คำคือคำว่า *ที่ การ เป็น ได้ จะ* และ *ใน* โดยที่คำแต่ละคำมีค่าความน่าจะเป็นที่จะเกิดในข้อมูลต้นกำเนิดเรียงลำดับจากมากไปหาน้อย

คำไทยทั้ง 6 คำ จะถูกเรียงลำดับตามความยาวของคำ โดยที่คำว่า *เป็น* จะถูกย้ายมาอยู่หน้าสุดของสายข้อมูล เนื่องจากมีจำนวนตัวอักษรในคำมากที่สุด คือ 4 ตัวอักษร สำหรับคำว่า *ที่ การ* และ *ได้* มีความยาวของคำเท่ากันหมดทั้ง 3 ตัว ฉะนั้น จึงต้องทำการเรียงลำดับตามค่าความน่าจะเป็นของคำที่เกิดขึ้นบ่อยจากมากไปหาน้อย เมื่อเรียงลำดับคำเรียบร้อยแล้ว ก็จะกำหนดรหัสแทนคำให้กับคำไทยแต่ละคำ จากตัวอย่างคำไทยทั้ง 6 จะใช้รหัสแทนคำ คือ 10 11 12 13 14 และ 15(รหัสแทนคำของคำว่า *ใน* เป็นคำสุดท้าย) ตามลำดับ เมื่อคำทั้ง 6 คำถูกกำหนดรหัสแทนคำให้เรียบร้อยแล้ว ลักษณะของสายข้อมูลในพจนานุกรมคำไทยจะเป็นดังนี้

[10]เป็น[11]ที่[12]การ[13]ได้[14]จะ[15]ใน

3.1.2 พจนานุกรม 2 ไบต์

พจนานุกรม 2 ไบต์ จะมีความแตกต่างจากพจนานุกรม 1 ไบต์ ตรงที่ตัวรหัสแทนคำของพจนานุกรมนี้ จะมีขนาด 2 ไบต์ เพื่อให้สามารถรองรับคำได้มากกว่า 109 คำ ซึ่งรหัสแทนคำขนาด 2 ไบต์นี้ จะเปรียบเสมือนตัวบอกตำแหน่งข้อมูลของโครงสร้างข้อมูลแบบอาร์เรย์สองมิติ(Two-

Dimensional Array) โดยไบต์แรกของรหัสแทนคำจะเปรียบเสมือนแถวของอาร์เรย์(Rows) และไบต์ที่สองของรหัสแทนคำจะเปรียบเสมือนหลักของอาร์เรย์(Columns) โดยแต่ละแถว จะมีหลักทั้งหมด 256 หลัก คือ ค่าที่เป็นไปได้ทั้งหมดของรหัสแอสกีขนาด 1 ไบต์ คือ 256 ค่า(2^8 ค่า) นั่นเองฉะนั้นรหัสที่สามารถอ้างอิงคำศัพท์ในพจนานุกรมจึงมีจำนวน 256 รหัสเช่นกัน แต่สำหรับจำนวนแถวที่เป็นไปได้ทั้งหมดของพจนานุกรมแบบนี้ จะมีเพียง 127 แถว เนื่องจากการเก็บข้อมูลในพจนานุกรมคำไทยนี้ รหัสแทนคำและคำไทยจะเรียงสลับต่อกันตลอดทั้งสายข้อมูล ฉะนั้นเพื่อให้สามารถแยกได้ว่ารหัสดังกล่าวเป็นรหัสแทนคำหรือรหัสแอสกีของอักษรไทย จึงกำหนดให้ไบต์แรกของรหัสแทนคำต้องไม่เป็นรหัสแอสกีที่อยู่ในกลุ่มอักษรไทย(7F ถึง FF) ซึ่งก็คือ จะต้องเป็นรหัสแอสกีที่อยู่ในช่วง 0 ถึง 7E(เลขฐาน 16) จำนวน 127 ค่า(ประมาณ 2^7)

จากข้อมูลข้างต้น ทำให้ทราบว่ารหัสแทนคำของพจนานุกรมนี้ จะมีค่าที่มีความเป็นหนึ่งเดียว 127 X 256 = 32,512 ค่า(ประมาณ 2^{15} ค่า) ซึ่งก็คือสามารถอ้างอิงคำไทยในพจนานุกรมได้ประมาณ 32,000 คำ

การกำหนดค่ารหัสแทนคำให้กับคำไทยแต่ละคำ จะไล่ลำดับจากน้อยไปหามากเช่นเดิม โดยไบต์แรกจะไล่ลำดับค่ารหัสแอสกีในรูปของเลขฐาน 16 ตั้งแต่ 0 ถึง 7E และไบต์ที่สองจะมีไล่ลำดับค่ารหัสแอสกีในรูปของเลขฐาน 16 ตั้งแต่ 0 ถึง FF ดังเช่นตัวอย่างที่ 3.2

<u>ตัวอย่างที่ 3.2</u> สมมติว่าจะเก็บคำไทยลงในพจนานุกรมคำไทยที่ใช้รหัสแทนคำ 2 ไบต์ จำนวน 5 คำคือคำว่า *ส่วน บาง ใหญ่ เอา* และ *เลย* โดยที่คำแต่ละคำมีค่าความน่าจะเป็นที่จะเกิดในข้อมูลต้นกำเนิดเรียงลำดับจากมากไปหาน้อย

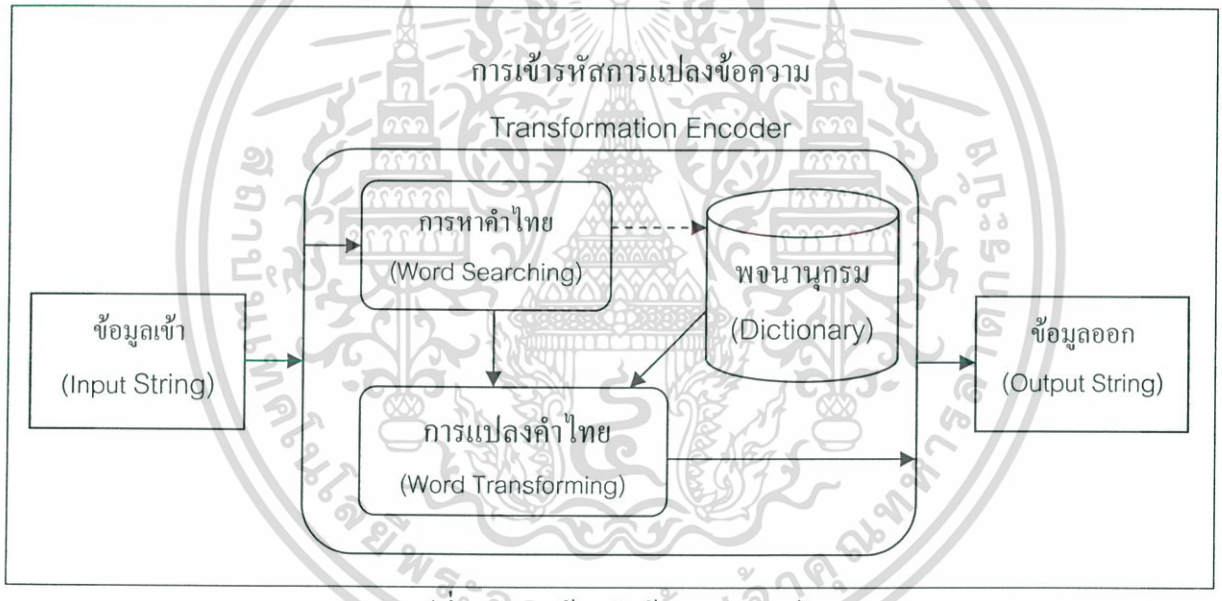
คำไทยทั้ง 5 คำ จะถูกเรียงลำดับตามความยาวของคำ โดยที่คำว่า *ใหญ่* จะถูกย้ายมาเป็นคำที่ 2 ของสายข้อมูล เนื่องจากว่ามีจำนวนตัวอักษรในคำ คือ 4 ตัวอักษร เท่ากับคำว่า *ส่วน* แต่ก็ยังมีค่าความน่าจะเป็นน้อยกว่าคำว่า *ส่วน* สำหรับคำว่า *บาง เอา* และ *เลย* มีความยาวของคำเท่ากันหมดทั้ง 3 ตัว ฉะนั้น จึงต้องทำการเรียงลำดับตามค่าความน่าจะเป็นของคำที่เกิดขึ้นบ่อยจากมากไปหาน้อย จากในตัวอย่างมีคำไทยทั้งหมด 5 คำ รหัสแทนคำที่ใช้ในไบต์แรกของทุกคำจะเป็น 0 และรหัสแทนคำที่ใช้ในไบต์ที่สองจะเป็น 0 1 2 3 และ 4 ตามลำดับ เมื่อคำทั้ง 5 คำ ถูกกำหนดรหัสแทนคำให้เรียบร้อยแล้ว ลักษณะของสายข้อมูลในพจนานุกรมคำไทยจะเป็นดังนี้

[0][0]ส่วน[0][1]ใหญ่[0][2]บาง[0][3]เอง[0][4]เลย

จากตัวอย่าง [0][0] เป็นรหัสแทนคำที่มีขนาด 2 ไบต์ ของคำว่า “ส่วน” [0][1] เป็นรหัสแทนคำขนาด 2 ไบต์ ของคำว่า “บาง” ตามลำดับ

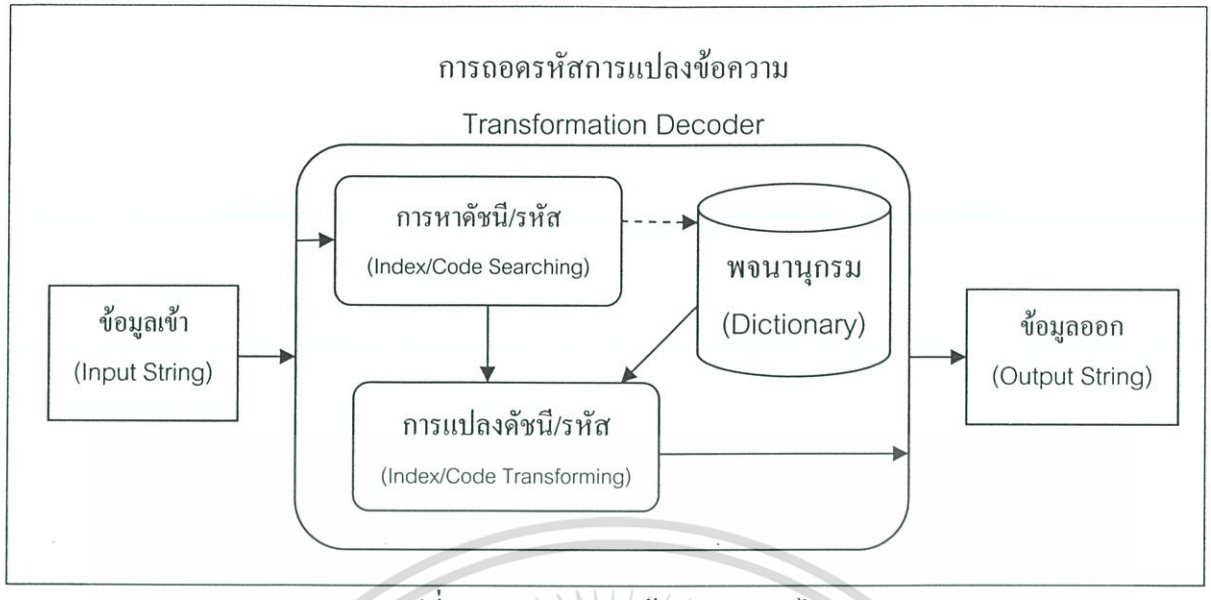
3.2 การแปลงข้อความภาษาไทย

ในหัวข้อการแปลงข้อความภาษาไทยนี้ จะกล่าวถึงขั้นตอนการเข้ารหัสและถอดรหัสข้อความภาษาไทย ที่ใช้เทคนิคพจนานุกรมคำศัพท์ในการแปลงข้อมูล โดยหลักการพื้นฐานของการเข้ารหัสข้อความภาษาไทย คือ การเข้ารหัสการแปลงข้อความ จะมีตัวโปรแกรมที่เรียกว่าตัวเข้ารหัสการแปลงข้อความ(Transformation Encoder) ซึ่งมีหน้าที่ทำการอ่านข้อมูลจากข้อมูลต้นกำเนิดทีละไบต์ แล้วนำเอามาตรวจสอบกับคำในพจนานุกรมคำไทยที่จัดเตรียมไว้ว่ามีคำใดตรงกันกับคำในพจนานุกรมหรือไม่ ถ้าหากพบคำในข้อมูลต้นกำเนิดตรงกับคำที่อยู่พจนานุกรมคำไทย ก็จะนำเอารหัสแทนคำของคำดังกล่าวมาวางแทนที่คำในข้อมูลต้นกำเนิด ดังรูปที่ 3.1



รูปที่ 3.1 การเข้ารหัสข้อความภาษาไทย

สำหรับกรณีของการถอดรหัสการแปลงข้อความภาษาไทยจะมีตัวถอดรหัสการแปลงข้อความ(Transformation Decoder) เช่นกัน ดังรูปที่ 3.2



รูปที่ 3.2 การถอดรหัสข้อความภาษาไทย

ตัวถอดรหัสการแปลงข้อความ จะทำหน้าที่อ่านข้อมูลทีละไบต์ เพื่อถอดรหัสแทนคำที่อยู่ในข้อมูลที่แปลงแล้ว ให้กลับมาอยู่ในสภาพของข้อมูลต้นกำเนิดดังเดิม ฉะนั้นเพื่อความถูกต้องในการเข้ารหัสและถอดรหัสการแปลงข้อความภาษาไทย พจนานุกรมคำไทยที่ใช้สำหรับการเข้ารหัสและถอดรหัสการแปลงข้อความภาษาไทยจะต้องเป็นพจนานุกรมเดียวกัน

การเข้ารหัสและถอดรหัสการแปลงข้อความภาษาไทย ตัวเข้ารหัสและถอดรหัสจะนำเอาข้อมูลทีละไบต์ที่อ่านได้ มาสืบค้นคำไทยในพจนานุกรมด้วยการค้นหาข้อมูลเชิงเส้น(Linear Search) ฉะนั้นเพื่อให้ค้นหาคำในพจนานุกรมได้ถูกต้อง พจนานุกรมคำไทยที่ใช้ในการอ้างอิงจึงต้องเรียงลำดับตามจำนวนอักษรของคำจากมากไปหาน้อยเพื่อป้องกันไม่ให้เกิดคำนำหน้า ดังตัวอย่างที่ 3.3

ตัวอย่างที่ 3.3 สมมติว่าในพจนานุกรมคำไทยมี คำที่บรรจุอยู่ในพจนานุกรม 2 คำ คือคำว่า *อย่า* และ *อย่าง* และมีรหัสแทนคำคือ *10* และ *11* ตามลำดับ พจนานุกรมคำไทยจะมีลักษณะดังนี้

[10]อย่า [11]อย่าง

เมื่อกำหนดให้ข้อมูลต้นกำเนิดมีข้อความว่า *อย่างนี้นี่เอง* ตัวเข้ารหัสการแปลงข้อความภาษาไทย จะอ่านข้อมูลทีละไบต์ แล้วนำตรวจสอบกับคำในพจนานุกรม ซึ่งเมื่อเข้ารหัสข้อมูลเรียบร้อยแล้ว ข้อมูลที่แปลงแล้วจะมีลักษณะดังนี้

[10]*งนี้นี่เอง*

จะเห็นได้อย่างชัดเจนว่า คำว่า *อย่าง* จะไม่มีทางถูกนำมาใช้ในการเข้ารหัสเลย ฉะนั้นการต้องเรียงลำดับคำไทยในพจนานุกรม ต้องให้คำที่มีความยาวมากอยู่หน้าคำที่มีความยาวน้อย โดยในตัวอย่างนี้ จะต้องเรียงลำดับคำในพจนานุกรมใหม่ คือ ให้คำว่า *อย่าง* อยู่หน้าคำว่า *อย่า* และรหัสแทนคำ คือ *10* และ*11* ตามลำดับ นั่นเอง

3.2.1 การแปลงข้อความด้วยพจนานุกรม 2 ไบต์

การแปลงข้อความภาษาไทยวิธีนี้ จะใช้พจนานุกรม 2 ไบต์ เป็นตัวอ้างอิงการเข้ารหัสและถอดรหัสข้อมูล โดยในการทดลอง ผู้ทำวิทยานิพนธ์กำหนดให้จำนวนคำไทยในพจนานุกรมมีจำนวนแตกต่างกันใน 2 ลักษณะ คือ จำนวน 511 คำ(คำไทยทั้งหมดจากตารางที่ ก1 ภาคผนวก ก) และ จำนวน 255 คำ ซึ่งจะเรียกวิธีการแปลงข้อความภาษาไทยที่ใช้พจนานุกรมที่เก็บคำ 511 คำ และ 255 คำ ว่า การแปลงข้อความด้วยคำจากสถิติทั้งหมด และการแปลงข้อความด้วยคำจากสถิติ 255 คำ ตามลำดับ

3.2.1.1 การแปลงข้อความด้วยคำจากสถิติทั้งหมด

การแปลงข้อความด้วยคำจากสถิติทั้งหมด จะเอาคำไทยจากตารางที่ ก1 ภาคผนวก ก ทั้งหมด 511 คำ เก็บลงในพจนานุกรมที่ใช้แปลงข้อมูล ซึ่งจากการวิเคราะห์คำไทยทั้ง 511 คำ พบว่าค่าความน่าจะเป็นที่จะเกิดคำทั้ง 511 คำ มีค่ามากกว่า 70 เปอร์เซ็นต์ มีค่าความยาวเฉลี่ยของคำ และมีค่าประมาณ 3.6 ตัวอักษรต่อคำ ฉะนั้นเพื่อให้การเข้ารหัสการแปลงข้อความภาษาไทยมีส่วนช่วยลดขนาดข้อมูล จึงกำหนดให้ตัวเข้ารหัสการแปลงข้อความภาษาไทยนี้ เข้ารหัสคำในข้อมูลต้นกำเนิดให้มีขนาดน้อยกว่า 3.6 ไบต์ ซึ่งในการอ้างอิงคำไทยในพจนานุกรมทั้ง 511 คำ จำเป็นต้องใช้รหัสข้อมูลที่มีขนาดอย่างน้อย 2 ไบต์ และในการเข้ารหัสคำไทยจะใช้รหัสบอกตำแหน่งการเข้ารหัสอีก 1 ไบต์ ดังนั้นในการแปลงคำไทย 1 คำ จึงใช้พื้นที่ขนาด 3 ไบต์ ซึ่งจะทำให้ข้อมูลลดลงเฉลี่ย 0.6 ไบต์ต่อคำ โดยในลำดับต่อไป จะกล่าวถึงการเข้ารหัสและถอดรหัสของวิธีการแปลงข้อความด้วยคำจากสถิติทั้งหมด โดยจะเรียกการเข้ารหัสและถอดรหัสดังกล่าวว่า การเข้ารหัสข้อความด้วยคำจากสถิติทั้งหมด และการถอดรหัสข้อความด้วยคำจากสถิติทั้งหมด ตามลำดับ

1 การเข้ารหัสข้อความด้วยคำจากสถิติทั้งหมด

การเข้ารหัสข้อความด้วยคำจากสถิติทั้งหมด จะเข้ารหัสคำไทยในข้อมูลต้นกำเนิดที่มีความหมายตรงกับคำไทยในพจนานุกรมที่จัดเตรียมไว้ โดยแปลงคำในข้อมูลต้นกำเนิดดังกล่าว ให้เป็นรหัสที่มีส่วนประกอบ 2 ส่วน คือส่วนดัชนีและส่วนรหัสแทนคำ โดยส่วนดัชนีจะใช้สัญลักษณ์ ' * ' (ขนาด 1 ไบต์) มีหน้าที่เป็นตัวบอกตำแหน่งเริ่มต้นการเข้ารหัสข้อมูลและส่วนรหัสแทนคำ (ขนาด 2 ไบต์) มีหน้าที่บอกตำแหน่งของคำดังกล่าวในพจนานุกรม ซึ่งขั้นตอนการเข้ารหัสข้อความจะมีการทำงาน 2 ขั้นตอน คือ ขั้นตอนการหาคำจากพจนานุกรมคำไทย และขั้นตอนการแปลงคำไทย

1) การหาคำจากพจนานุกรมคำไทย

ขั้นตอนนี้จะทำการอ่านคำจากข้อมูลต้นกำเนิดทีละไบต์ เพื่อนำมาเปรียบเทียบว่ามีคำใดอยู่ในพจนานุกรมคำไทยที่เตรียมไว้หรือไม่ หากพบคำที่ตรงกับพจนานุกรมคำไทยก็จะทำการส่งค่าของรหัสแทนคำนั้น ให้กับส่วนการแปลงคำไทยต่อไป หากพบตัวอักษรหรือคำที่ไม่มีอยู่ในพจนานุกรมคำไทยก็จะทำการเขียนข้อมูลลงไปในข้อมูลที่แปลงแล้วโดยทันที ดังตัวอย่างที่ 3.4

ตัวอย่างที่ 3.4

ข้อมูลต้นกำเนิด : อาตี๋ใหญ่ไม่มีเงินไปเที่ยวเป็นอาทิตย์แล้ว

พจนานุกรมคำไทย : #@ด้วย#Aเป็น#Bที่#Cการ#Dได้#Eไม่#Fมี#Gจะ

ข้อมูลขณะทำการค้นหา : อาตี๋ใหญ่ไม่

จากตัวอย่างที่ 3.4 ตัวเข้ารหัสการแปลงข้อความภาษาไทยจะทำการอ่านคำจากข้อมูลต้นกำเนิดทีละไบต์ แล้วนำมาเปรียบเทียบว่ามีคำใดอยู่ในพจนานุกรมคำไทยที่เตรียมไว้ ซึ่งจากตัวอย่างข้อมูลต้นกำเนิด จะทำการเขียนข้อความ "*อาตี๋ใหญ่*" ลงในข้อมูลที่แปลงแล้วโดยทันที จนถึงคำว่า "*ไม่*"(ที่ขีดเส้นใต้ไว้) คือ คำที่พบในพจนานุกรมคำไทยและมีรหัสแทนคำ คือ #E ในพจนานุกรมคำไทย จากนั้นก็จะส่งค่า #E ให้กับส่วนการแปลงคำไทยต่อไป

2) การแปลงคำไทย

ในขั้นตอนการแปลงคำไทย จะทำการแปลงคำไทยจากข้อมูลต้นกำเนิดให้อยู่ในโครงสร้างที่ใช้ ' * ' และใช้รหัสแทนคำที่ได้จากขั้นตอนการหาคำจากพจนานุกรมคำไทย ซึ่งคำในข้อมูลต้นกำเนิดจะถูกแทนที่ด้วยดัชนีและรหัสแทนคำ ดังรูปแบบต่อไปนี้

* [รหัสแทนคำ]

ในกรณีที่มีการเข้ารหัสคำในข้อมูลต้นกำเนิดในปริมาณมาก จะเกิดการซ้ำของตัวดัชนี ' * ' เพิ่มขึ้น ซึ่งส่งผลดีต่อการบีบอัดข้อมูลที่อาศัยความซ้ำซ้อนของตัวอักษร และเมื่อทำการแทนที่คำไทยในข้อมูลต้นกำเนิดด้วยรหัสแทนคำที่ได้จากพจนานุกรมคำไทยเรียบร้อยแล้ว ตัวเข้ารหัสข้อมูลก็จะกลับไปค้นหาคำจากพจนานุกรมคำไทยเช่นเดียวกันกับขั้นตอนการหาคำจากพจนานุกรมคำไทยเช่นเดิมจนกว่าข้อมูลจะหมด ดังแสดงในตัวอย่างที่ 3.5

ตัวอย่างที่ 3.5

ข้อมูลต้นกำเนิด : *อาตี๋ใหญ่ไม่มีเงินไปเที่ยวเป็นอาทิตย์แล้ว*

พจนานุกรมคำไทย : #@ด้วย#Aเป็น#Bที่#Cการ#Dได้#Eไม่#Fมี#Gจะ

ข้อมูลที่แปลงแล้ว : *อาตี๋ใหญ่*#E*

ตัวอย่างที่ 3.5 แสดงการนำรหัสแทนคำที่ได้จากขั้นตอนการหาคำจากพจนานุกรมคำไทยในตัวอย่างที่ 3.4 คือ #E(ที่ขีดเส้นใต้ไว้) มาทำการเข้ารหัสการแปลงข้อความภาษาไทยให้อยู่ในรูปของตัวดัชนีและรหัสแทนคำ คือ *#E(ที่ขีดเส้นใต้ไว้)

เมื่อเข้ารหัสข้อมูลต้นกำเนิดจาก ตัวอย่างที่ 3.4 และ ตัวอย่างที่ 3.5 เรียบร้อยแล้ว ข้อมูลที่แปลงแล้วจะมีลักษณะดังข้อความต่อไปนี้

ข้อมูลที่แปลงแล้ว : *อาตี๋ใหญ่*#E*#Fเงินไปเที่ยว*#Aอาทิตย์แล้ว*

2 การถอดรหัสข้อความด้วยคำจากสถิติทั้งหมด

การถอดรหัสข้อความด้วยคำจากสถิติทั้งหมด คือ การถอดรหัสการแปลงข้อความในลักษณะทำงานย้อนกลับการเข้ารหัสข้อความด้วยคำจากสถิติทั้งหมด โดยขั้นตอนการทำงานจะแบ่งออกเป็น 2 ขั้นตอน คือ ขั้นตอนการหาดัชนีจากข้อมูลที่แปลงแล้ว และ ขั้นตอนการแปลงดัชนีและรหัสแทนคำ

1) การหาดัชนีจากข้อมูลที่แปลงแล้ว

ขั้นตอนนี้จะทำการตรวจสอบตัวอักษรจากข้อมูลที่แปลงแล้วทีละตัว หากพบตัวอักษร ‘ * ’ ก็จะทำการอ่านค่ารหัสที่ตามหลังตัว ‘ * ’ ขนาด 2 ไบต์ แล้วนำไปรหัสที่ได้ดังกล่าว ไปผ่านขั้นตอนการแปลงดัชนีและรหัสแทนคำเป็นคำไทย ในกรณีที่ไม่พบตัวอักษร ‘ * ’ ก็จะทำการเขียนอักษรนั้นลงไปในข้อมูลต้นกำเนิดทันที ดังตัวอย่างที่ 3.6

ตัวอย่างที่ 3.6

ข้อมูลที่แปลงแล้ว : *อาตี๋ใหญ่*#E*#Fเงินไปเที่ยว*#Aอาทิตย์แล้ว*

↓

พจนานุกรมคำไทย : #@ด้วย#Aเป็น#Bที่#Cการ#Dได้#Eไม่#Fมี#Gจะ

↓

ข้อมูลต้นกำเนิด : *อาตี๋ใหญ่*#E*

จากตัวอย่างที่ 3.6 ตัวถอดรหัสจะอ่านข้อความจากข้อมูลต้นกำเนิดทีละไบต์ แล้วตรวจสอบว่าเป็นสัญลักษณ์ ‘ * ’ หรือไม่ หากไม่ใช่ก็จะทำการเขียนอักษรตัวนั้นกลับสู่ข้อมูลต้นกำเนิดทันที ดังเช่นข้อความ “*อาตี๋ใหญ่*” ในตัวอย่าง จะถูกเขียนลงสู่ข้อมูลต้นกำเนิดโดยไม่ต้องนำมาตรวจสอบกับพจนานุกรมคำไทย แต่เมื่อพบสัญลักษณ์ ‘ * ’ ก็จะทำการนำค่า #E(ที่ขีดเส้นใต้ไว้) ส่งให้กับขั้นตอนการแปลงดัชนีและรหัสแทนคำเป็นคำไทยต่อไป

2) การแปลงดัชนีและรหัสแทนคำเป็นคำไทย

เมื่อได้รับรหัสแทนคำขนาด 2 ไบต์จากขั้นตอนการหาดัชนีจากข้อมูลที่แปลงแล้ว จะนำเอาข้อมูลขนาด 2 ไบต์ที่ได้รับ ไปตรวจสอบกับรหัสแทนคำที่เก็บอยู่ในพจนานุกรมคำไทยที่เตรียมไว้ เมื่อพบรหัสแทนคำดังกล่าวในพจนานุกรม ก็จะนำคำที่อยู่หลังรหัสดังกล่าวมาเขียนลงในข้อมูลต้นกำเนิด เมื่อทำการแปลงดัชนีและรหัสแทนคำเป็นคำไทยเรียบร้อยแล้ว ตัวถอดรหัสก็จะทำกลับไปทำการหาดัชนีจากข้อมูลที่แปลงแล้วต่อไปจนกว่าข้อมูลจะหมด ดังตัวอย่างที่ 3.7

ตัวอย่างที่ 3.7

ข้อมูลที่แปลงแล้ว : *อาตี๋ใหญ่*#E*#Fเงินไปเที่ยว*#Aอาทิตย์แล้ว*

พจนานุกรมคำไทย : #@ด้วย#Aเป็น#Bที่#Cการ#Dได้#Eไม่#Fมี#Gจะ

ข้อมูลต้นกำเนิด : *อาตี๋ใหญ่ไม่มีเงินไปเที่ยวเป็นอาทิตย์แล้ว*

จากตัวอย่างที่ 3.7 คำว่า “*ไม่*” “*มี*” และ “*เป็น*” (ที่ขีดเส้นใต้ไว้) ในข้อมูลต้นกำเนิด เป็นคำที่ถูกแปลงจากรหัสแทนคำ #E #F และ #A ที่พบในข้อมูลที่แปลงแล้ว ตามลำดับ

3.2.1.2 การแปลงข้อความด้วยคำจากสถิติ 255 คำ

ถึงแม้ว่าการแปลงข้อความด้วยคำจากสถิติทั้งหมดจะสามารถทำให้คำในข้อมูลต้นกำเนิดที่ตรงกับพจนานุกรมมีขนาดลดลงเฉลี่ย 0.6 ไบต์ แต่ขนาดที่เล็กลงเฉลี่ยเพียง 0.6 ไบต์ต่อคำ อาจยังไม่เพียงพอที่จะทำให้การแปลงข้อมูลมีประสิทธิภาพสูงสุด ผู้ทำวิทยานิพนธ์จึงกำหนดให้เก็บคำในพจนานุกรมไม่เกิน 256 คำ คือ ใช้รหัสในการอ้างอิงเพียงแถวเดียวเท่านั้น(256 ค่า) หรือกล่าวอีกนัยหนึ่งคือ ไบต์แรกของรหัสแทนคำของพจนานุกรม 2 ไบต์ จะถูกกำหนดให้มีค่าเดียวกันตลอดทั้งพจนานุกรม ซึ่งทำให้ไม่จำเป็นต้องนำไปใช้ในการเข้ารหัสข้อมูล เพียงแต่มีหน้าที่เป็นตัวคั่นคำในสายข้อมูลเท่านั้น

พจนานุกรมคำไทยของการแปลงข้อมูลวิธีนี้ จะบรรจุคำไทย 255 คำแรกจากตารางที่ ก1 ในภาคผนวก ก ซึ่งค่าความน่าจะเป็นที่จะเกิดคำทั้ง 255 คำในข้อมูลต้นกำเนิด มีค่าประมาณ 66 เปอร์เซ็นต์ และมีค่าความยาวเฉลี่ยของคำประมาณ 3.4 ตัวอักษร ฉะนั้นเพื่อให้การเข้ารหัสการ

แปลงข้อความภาษาไทยมีส่วนช่วยลดขนาดข้อมูล จึงกำหนดให้ตัวเข้ารหัสการแปลงข้อความภาษาไทยนี้ เข้ารหัสคำในข้อมูลต้นกำเนิดให้มีขนาดน้อยกว่า 3.4 ไบต์ ซึ่งในการอ้างอิงคำไทยในพจนานุกรมทั้ง 255 คำ จำเป็นต้องใช้รหัสข้อมูลที่มีขนาดอย่างน้อย 1 ไบต์ และในการเข้ารหัสจำเป็นต้องใช้รหัสบอกตำแหน่งการเข้ารหัสอีก 1 ไบต์ ดังนั้นในการแปลงคำไทย 1 คำ จึงใช้พื้นที่ขนาด 2 ไบต์ ซึ่งจะทำให้ข้อมูลลดลงเฉลี่ย 1.4 ตัวอักษร

โดยในลำดับต่อไป จะกล่าวถึงการเข้ารหัสและถอดรหัสของวิธีการแปลงข้อความด้วยคำจากสถิติทั้งหมด ซึ่งจะเรียกการเข้ารหัสและถอดรหัสดังกล่าวว่า การเข้ารหัสข้อความด้วยคำจากสถิติทั้งหมด และการถอดรหัสข้อความด้วยคำจากสถิติทั้งหมด ตามลำดับ

1 การเข้ารหัสข้อความด้วยคำจากสถิติ 255 คำ

การเข้ารหัสข้อความด้วยคำจากสถิติ 255 คำ จะมีความแตกต่างการเข้ารหัสข้อความด้วยคำจากสถิติทั้งหมดเพียงเล็กน้อย โดยส่วนข้อมูลที่ใช้เข้ารหัส จะมี 2 ส่วนเช่นเดิม คือ ส่วนดัชนียังคงใช้สัญลักษณ์ ' * ' เป็นตัวบอกตำแหน่งเริ่มต้นการแปลงข้อมูล แต่อีกส่วนจะใช้ไบต์ที่สองของรหัสแทนคำ ในการอ้างอิงคำไทยในพจนานุกรมเท่านั้น ฉะนั้นในการเข้ารหัสคำแต่ละคำในข้อมูลต้นกำเนิดจะใช้พื้นที่ในเก็บเพียง 2 ไบต์ คือ ดัชนี 1 ไบต์ และไบต์ที่สองของรหัสแทนคำอีก 1 ไบต์

สาเหตุที่บรรจุคำไทยไว้ในพจนานุกรมเพียง 255 คำ แม้ว่าพจนานุกรมสามารถบรรจุคำไทยได้สูงสุด 256 คำ(2^8 คำ) เนื่องจากในข้อมูลต้นกำเนิดอาจมีตัว ' * ' ที่ตรงกับสัญลักษณ์ของดัชนี ส่งผลให้ข้อมูลที่แปลงแล้ว ไม่สามารถถอดรหัสกลับได้ ผู้ทำวิทยานิพนธ์จึงกำหนดให้ ' * ' เป็นคำหนึ่งคำในพจนานุกรมคำไทย และอีก 255 คำ ที่เหลือเป็นคำไทย

สำหรับขั้นตอนการเข้ารหัสข้อความด้วยคำจากสถิติ 255 คำ จะมี 2 ขั้นตอนเช่นเดียวกับการหาคำจากพจนานุกรมคำไทยและการแปลงคำไทยของการแปลงข้อความด้วยคำจากสถิติทั้งหมดในหัวข้อที่ 3.2.1.1 ดังแสดงในตัวอย่างที่ 3.8

ตัวอย่างที่ 3.8

ข้อมูลต้นกำเนิด : *ไม่มีใครอยากไปเที่ยวเป็นเพื่อนผมเลย*

↓

พจนานุกรมคำไทย : @ด้วยAเป็นBที่CการDได้Eไม่FมีGจะ

↓

ข้อมูลที่แปลงแล้ว : **E*Fใครอยากไปเที่ยว*Aเพื่อนผมเลย*

ตัวอย่างที่ 3.8 แสดงการเข้ารหัสการแปลงข้อความภาษาไทยในข้อมูลที่แปลงแล้ว คือ *E *F และ *A(ที่ขีดเส้นใต้) โดยที่ ' * ' ที่นำหน้าคือตัวดัชนี และตัว E F และ A คือตัวไบต์ที่สองของรหัสแทนในพจนานุกรมที่ได้จากการเข้ารหัสข้อมูลต้นกำเนิดที่มีคำไทย คือ "*ไม่*" "*มี*" และ "*เป็น*" จากพจนานุกรม ตามลำดับ

2 การถอดรหัสข้อความด้วยคำจากสถิติ 255 คำ

การเข้ารหัสข้อความด้วยคำจากสถิติ 255 คำ มีลักษณะการทำงานเช่นเดียวกับการถอดรหัสการแปลงข้อความภาษาไทยที่ใช้คำไทยที่ได้จากสถิติทั้งหมด ในหัวข้อที่ 3.2.1.1 แต่มีการเปลี่ยนแปลงขั้นตอนเพียงเล็กน้อย คือ ในการหาดัชนีจากข้อมูลที่แปลงแล้ว จะทำการอ่านค่ารหัสที่ตามหลังตัว ' * ' ขนาด 1 ไบต์ และการแปลงดัชนีและรหัสแทนคำเป็นคำไทย จะนำเอารหัสที่ได้จากขั้นตอนการหาดัชนีจากข้อมูลที่แปลงแล้วขนาด 1 ไบต์ ไปตรวจสอบกับไบต์ที่สองของรหัสแทนคำที่เก็บอยู่ในพจนานุกรมคำไทย เมื่อพบรหัสแทนคำดังกล่าวในพจนานุกรม ก็จะนำคำที่อยู่หลังรหัสดังกล่าวมาเขียนลงในข้อมูลต้นกำเนิด ดังตัวอย่างที่ 3.9

ตัวอย่างที่ 3.9

ข้อมูลที่แปลงแล้ว : *E*F*ใครอยาก*A*เพื่อนผมเลย

พจนานุกรมคำไทย : @ด้วยAเป็นBที่CการDได้Eไม่FมีGจะ

ข้อมูลต้นกำเนิด : *ไม่มี*ใครอยาก*ไปเที่ยว**เป็น**เพื่อน*ผมเลย

จากตัวอย่างที่ 3.9 คำว่า "*ไม่*" "*มี*" และ "*เป็น*" (ที่ขีดเส้นใต้ไว้) ในข้อมูลต้นกำเนิด เป็นคำที่ถอดรหัส *E *F และ *A ที่พบในข้อมูลที่แปลงแล้ว ตามลำดับ

3.2.2 การแปลงข้อความด้วยพจนานุกรม 1 ไบต์

แม้ว่าในการแปลงข้อความด้วยคำจากสถิติ 255 คำ จะสามารถทำให้คำในข้อมูลต้นกำเนิดที่ตรงกับพจนานุกรมมีขนาดลดลงเฉลี่ย 1.4 ไบต์ แต่ขนาดที่เล็กลงเฉลี่ย 1.4 ไบต์ต่อคำ อาจยังไม่เพียงพอที่จะทำให้การแปลงข้อมูลมีประสิทธิภาพสูงสุด ฉะนั้นเพื่อให้ข้อมูลหลังเข้ารหัสมีขนาดเล็ก

ที่สุด การแปลงข้อความวิธีนี้ จึงจะไม่ใช้ตัวดัชนี ' * ' ในการบอกตำแหน่งที่มีการแปลงข้อมูล แต่จะใช้เทคนิคบางอย่างเพื่อให้สามารถถอดรหัสข้อมูลได้ ซึ่งจะกล่าวต่อไป

จากหัวข้อที่ 3.1.1 พจนานุกรมที่ใช้ในการแปลงข้อมูลวิธีนี้ จะมีจำนวนคำในพจนานุกรมทั้งสิ้น 109 คำ(คำไทยทั้งหมดจากตารางที่ ก1 ภาคผนวก ก) ซึ่งจากการวิเคราะห์คำไทยทั้ง 109 คำ ค่าความน่าจะเป็นที่จะเกิดคำทั้งหมดในข้อมูลต้นกำเนิดประมาณ 50 เปอร์เซ็นต์ และมีค่าความยาวเฉลี่ยของคำประมาณ 3.2 ตัวอักษร ซึ่งหากทำการเข้ารหัสคำไทย 1 คำ ให้มีขนาดเหลือเพียง 1 ไบต์ จะทำให้คำดังกล่าวมีขนาดลดลงเฉลี่ย 2.2 ไบต์ โดย 1 ไบต์ดังกล่าวจะในการใช้อ้างอิงคำศัพท์ในพจนานุกรมเท่านั้น

จากที่กล่าวไว้ข้างต้น ผู้วิจัยเลือกใช้รหัส 7E(ค่ารหัสแอสกีของสัญลักษณ์ ' ~ ') เป็นตัวบอกตำแหน่งการเกิดรหัสที่มีค่าเดียวกันกับรหัสที่ใช้ในการแปลงข้อมูล(รหัสที่อยู่ในกลุ่มของตัวอักษรภาษาอังกฤษ และตัวอักษรที่ใช้ในระบบการสื่อสารข้อมูล) อยู่หลังรหัส 7E จะเป็นข้อความที่ไม่มีการแปลงข้อมูลเกิดขึ้น โดยจะกล่าวถึงวิธีการเข้ารหัสและถอดรหัสการแปลงข้อความด้วยพจนานุกรม 1 ไบต์ เป็นลำดับต่อไป

3.2.2.1 การเข้ารหัสข้อความด้วยพจนานุกรม 1 ไบต์

การเข้ารหัสข้อความด้วยพจนานุกรม 1 ไบต์ คือ การเข้ารหัสการแปลงข้อความภาษาไทยที่ไม่มีการใช้ดัชนี แต่จะใช้รหัสแทนคำที่ได้จากพจนานุกรม 1 ไบต์ แทนที่คำในข้อมูลต้นกำเนิด ซึ่งมีขั้นตอนการทำงานของการเข้ารหัสข้อความภาษาไทย 2 ส่วนเช่นเดิม คือ ขั้นตอนการหาคำจากพจนานุกรมคำไทย และขั้นตอนการแปลงคำไทย

1 การหาคำจากพจนานุกรมคำไทย

ขั้นตอนนี้ จะทำการอ่านคำจากข้อมูลต้นกำเนิด เพื่อนำมาเปรียบเทียบว่ามีคำใดอยู่ในพจนานุกรมคำไทยหรือไม่ หากพบคำที่ตรงกับพจนานุกรมคำไทยในช่วงคำที่ 1 ถึงคำที่ 109 ก็จะนำเอารหัสแทนคำจากพจนานุกรม(1 ไบต์) ส่งให้ขั้นตอนการแปลงคำไทยต่อไป ในกรณีที่พบอักษรที่มีรหัสแอสกีในข้อมูลต้นกำเนิดในช่วง 10 ถึง 7D ยกเว้น ตัวเว้นวรรค(20) จะนำรหัสแอสกีที่มีค่า 7E(รหัสแอสกีของสัญลักษณ์ ' ~ ') ใส่ไว้ตอนต้นและตอนท้ายของอักษรหรือกลุ่มอักษรนั้น ดังตัว-อย่างที่ 3.10

ตัวอย่างที่ 3.10

เต่า Geochelone Sulcata เป็นเต่าที่มีขนาดใหญ่เป็นอันดับที่ 2 ของโลก

เต่า ~Geochelone Sulcata ~เป็นเต่าที่มีขนาดใหญ่เป็นอันดับที่ 2 ของโลก

จากตัวอย่างที่ 3.10 คำว่า "Geochelone Sulcata " ทั้งหมดเป็นประโยคภาษาอังกฤษที่มีค่าในรหัสแอสกีในช่วง 0 ถึง 7E จึงต้องทำการใส่ตัว ' ~ ' ไว้ด้านหน้าและด้านท้ายประโยคดังกล่าว สำหรับคำหรืออักษรอื่นที่ไม่มีอยู่ในพจนานุกรม และไม่ได้อยู่ในช่วง 10 ถึง 7E ยกเว้น ตัวเว้นวรรค (20) จะถูกนำมาเขียนลงแฟ้มข้อมูลที่แปลงแล้วโดยทันที

2 การแปลงคำไทย

เมื่อได้รับรหัสแทนคำจากขั้นตอนการหาคำจากพจนานุกรมคำไทย ก็จะนำรหัสแทนคำดังกล่าวมาทำการเข้ารหัสในลักษณะเดียวกับการแปลงคำไทยในหัวข้อที่ 3.2.1.2 แต่จะไม่มีการใส่ตัวดัชนี ' * ' ลงไปในตัวข้อมูลที่แปลงแล้ว ดังแสดงในตัวอย่างที่ 3.11

ตัวอย่างที่ 3.11

ข้อมูลต้นกำเนิด : *ABAC เป็นมหาวิทยาลัยที่มีชื่อเสียง*

พจนานุกรมคำไทย : @ที่AการBเป็นCได้DจะEด้วยFมีGไม่

ข้อมูลที่แปลงแล้ว : *~ABAC ~*Bมหาวิทยาลัย@Fชื่อเสียง

ตัวอย่างที่ 3.11 แสดงการนำรหัสแทนคำที่ได้จากขั้นตอนการหาคำจากพจนานุกรมคำไทย ได้แก่ คำว่า "เป็น" "ที่" และ "มี" (ที่ขีดเส้นใต้ไว้) มาทำการเข้ารหัสให้อยู่ในรูปของรหัสแทนคำ คือ B, @ และ F ตามลำดับ กลุ่มอักษรภาษาอังกฤษที่ใช้เป็นรหัสแทนคำในข้อมูลต้นกำเนิด คือ ABAC จะถูกคั่นด้วยรหัสแอสกีที่มีค่า 7E(รหัสแอสกีของสัญลักษณ์ ' ~ ') ทางด้านหน้าและด้านหลัง สำหรับข้อความ "มหาวิทยาลัย" และ "ชื่อเสียง" จะถูกเขียนลงในข้อมูลที่แปลงแล้ว โดยไม่ทำอะไร

3.2.2.2 การถอดรหัสข้อความด้วยพจนานุกรม 1 ไบต์

การถอดข้อความด้วยคำจากสถิติ 109 คำ คือ การถอดรหัสข้อมูลในลักษณะทำงานย้อนกลับการเข้ารหัสการแปลงข้อมูล โดยขั้นตอนการทำงานจะแบ่งออกเป็น 2 ขั้นตอนเช่นเดิม คือ ขั้น-ตอนการหารหัสจากข้อมูลที่แปลงแล้ว และขั้นตอนการถอดรหัสแทนคำเป็นคำไทย

1 การหารหัสจากข้อมูลที่แปลงแล้ว

ในขั้นตอนนี้จะทำการอ่านคำจากข้อมูลที่ต้องการจะแปลงกลับทีละตัว ในกรณีที่พบรหัสแทนคำ(พบในช่วง 10 ถึง 7D ยกเว้น ตัวเว้นวรรค) จะทำการตรวจสอบรหัสดังกล่าวในพจนานุกรมคำไทย หากพบรหัสที่ตรงกับพจนานุกรมคำไทยในระหว่างคำที่ 1 ถึงคำที่ 109 ก็จะนำคำที่อยู่หลังรหัสดังกล่าวในพจนานุกรมคำไทยส่งไปให้ขั้นตอนการแปลงรหัสแทนคำเป็นคำไทยต่อไป ในกรณีที่พบตัว ‘ ~ ’ ในข้อมูลที่แปลงแล้ว จะไม่ทำการพิจารณาอักษรหรือกลุ่มอักษรดังกล่าวที่อยู่หลังตัว ‘ ~ ’ โดยจะเขียนอักษรหรือกลุ่มอักษรดังกล่าวลงบนข้อมูลต้นกำเนิดทันที จนกว่าจะพบ ‘ ~ ’ อีกครั้งหนึ่ง สำหรับคำหรืออักษรอื่นที่ไม่มีอยู่ในพจนานุกรม และไม่ได้อยู่ในช่วง 10 ถึง 7D ยกเว้น ตัวเว้นวรรค(20) จะถูกนำมาเขียนลงแฟ้มข้อมูลต้นกำเนิดโดยทันที

2 การแปลงรหัสแทนคำเป็นคำไทย

เมื่อได้รับรหัสขนาด 1 ไบต์จากขั้นตอนการหาดัชนีจากข้อมูลที่แปลงแล้ว จะนำเอาข้อมูลขนาด 1 ไบต์ที่ได้รับ ไปตรวจสอบกับรหัสแทนคำที่เก็บอยู่ในพจนานุกรมคำไทยที่เตรียมไว้ เมื่อพบรหัสแทนคำดังกล่าวในพจนานุกรม ก็จะนำคำที่อยู่หลังรหัสดังกล่าวมาเขียนลงในข้อมูลต้นกำเนิด

3.3 สรุป

ในการพัฒนาวิธีการแปลงข้อความภาษาไทย ผู้วิจัยได้พัฒนาโปรแกรมสำหรับวิธีการแปลงข้อความภาษาไทยขึ้น 3 วิธี คือ วิธีการแปลงข้อความภาษาไทยด้วยคำจากสถิติทั้งหมด วิธีการแปลงข้อความด้วยคำจากสถิติ 255 คำ และวิธีการแปลงข้อความด้วยพจนานุกรม 1 ไบต์ แต่ละวิธีมีความแตกต่างกัน โดยสามารถได้ดังตารางที่ 3.1

ตารางที่ 3.1 สรุปความแตกต่างของวิธีการแปลงข้อความภาษาไทยแต่ละวิธี

การแปลงข้อความภาษาไทย	ด้วยพจนานุกรม 2 ไบต์		ด้วยพจนานุกรม 1 ไบต์
	ด้วยคำจากสถิติทั้งหมด	ด้วยคำจากสถิติ 255 คำ	
จำนวนคำในพจนานุกรม	511 คำ	255 คำ	109 คำ
หน่วยพื้นที่ ที่ใช้เข้ารหัสคำไทย 1 คำ	3 ไบต์	2 ไบต์	1 ไบต์
การเพิ่มจำนวนคำในพจนานุกรม	ทำได้	ทำไม่ได้	ทำไม่ได้
ช่วยลดขนาดข้อมูลเฉลี่ยต่อคำ	0.6 ไบต์	1.4 ไบต์	2.2 ไบต์

สำหรับการทดสอบประสิทธิภาพของการแปลงข้อความภาษาไทยทั้ง 3 วิธี จะกล่าวในบทต่อไป

บทที่ 4
การทดสอบประสิทธิภาพการแปลงข้อมูล

ในบทนี้จะเสนอการทดสอบประสิทธิภาพการแปลงข้อความภาษาไทยที่กล่าวไว้ในบทที่ 3 ทั้ง 3 วิธี โดยทำการพัฒนาโปรแกรมต้นแบบที่ใช้สำหรับเข้ารหัสและถอดรหัสข้อความภาษาไทย แล้วนำไปทดสอบกับข้อมูลที่จัดเตรียมไว้

เพื่อความสะดวกในการอ้างถึงทั้ง 3 วิธี จึงได้กำหนดชื่อย่อแต่ละวิธีไว้ดังนี้
วิธีการแปลงข้อความด้วยคำจากสถิติทั้งหมด เรียกว่า TTT3
วิธีการแปลงข้อความด้วยคำจากสถิติ 255 คำ เรียกว่า TTT2
วิธีการแปลงข้อความด้วยพจนานุกรม 1 ไบต์ เรียกว่า TTT1

4.1 ข้อมูลที่ใช้ในการทดสอบ

ข้อมูลที่ใช้ในการทดสอบเป็นข้อมูลที่สุ่มตัวอย่างจาก วารสาร นิตยสาร จดหมายราชการ รายงาน หนังสือพิมพ์ รวมทั้งข้อความที่ได้รับจากจดหมายอิเล็กทรอนิกส์ จำนวนทั้งสิ้น 100 ตัวอย่าง ซึ่งข้อมูลทั้งหมดเป็นข้อความภาษาไทยที่มีความทันสมัย และใช้กันอยู่ในปัจจุบัน โดยที่ภายในเนื้อหาข้อมูลอาจมีสัญลักษณ์ ตัวเลข ตัวอักษร กลุ่มวลี หรือข้อความภาษาอังกฤษปะปนบ้างเล็กน้อย

นำตัวอย่างทั้งหมดมาทำการจัดเรียงตามปริมาณข้อมูลจากน้อยไปมาก ดังตารางที่ 4.1 โดยแบ่งข้อมูลทั้งหมดออกเป็น 10 กลุ่มดังนี้

1)	ข้อมูลที่มีปริมาณน้อยกว่า 2 กิโลไบต์	10	ตัวอย่าง
	โดยแบ่งเป็น 2 กลุ่มย่อย คือ		
	- ข้อมูลที่มีปริมาณน้อยกว่า 1 กิโลไบต์	5	ตัวอย่าง
	- ข้อมูลที่มีปริมาณระหว่าง 1 กิโลไบต์ ถึง 2 กิโลไบต์	5	ตัวอย่าง
2)	ข้อมูลที่มีปริมาณระหว่าง 2 กิโลไบต์ ถึง 3 กิโลไบต์	10	ตัวอย่าง
3)	ข้อมูลที่มีปริมาณระหว่าง 3 กิโลไบต์ ถึง 4 กิโลไบต์	10	ตัวอย่าง
4)	ข้อมูลที่มีปริมาณระหว่าง 4 กิโลไบต์ ถึง 5 กิโลไบต์	10	ตัวอย่าง
5)	ข้อมูลที่มีปริมาณระหว่าง 5 กิโลไบต์ ถึง 6 กิโลไบต์	10	ตัวอย่าง

6) ข้อมูลที่มีปริมาณระหว่าง 6 กิโลไบต์ ถึง 7 กิโลไบต์ 10 ตัวอย่าง
7) ข้อมูลที่มีปริมาณระหว่าง 7 กิโลไบต์ ถึง 8 กิโลไบต์ 10 ตัวอย่าง
8) ข้อมูลที่มีปริมาณระหว่าง 8 กิโลไบต์ ถึง 9 กิโลไบต์ 10 ตัวอย่าง
9) ข้อมูลที่มีปริมาณระหว่าง 9 กิโลไบต์ ถึง 10 กิโลไบต์ 10 ตัวอย่าง
10) ข้อมูลที่มีปริมาณมากกว่า 10 กิโลไบต์ 10 ตัวอย่าง

ตารางที่ 4.1 ข้อมูลภาษาไทยที่ใช้ในการทดสอบ

ชื่อแฟ้มข้อมูล	ปริมาณข้อมูล(หน่วยเป็นไบต์)	รายละเอียด
a1.txt	854	อย่าคิดที่จะเปลี่ยน...
a2.txt	916	ก่อนที่จะรัก...
a3.txt	921	คือรักแท้...
a4.txt	925	ชีวิตเรา คือของ...
a5.txt	948	คือรักแท้...
a6.txt	1051	ขอเพียงความเข้าใจ...
a7.txt	1194	ข้อปฏิบัติและขั้น...
a8.txt	1262	การปล่อยวาง...
a9.txt	1737	ริมฝั่งแม่น้ำเซน...
a10.txt	1980	ใบหูที่หายไป...
b1.txt	2085	ตั้งสถาบันทดสอบ...
b2.txt	2149	กทม.ไม่สนับสนุน...
b3.txt	2167	ข้อแนะนำเกี่ยวกับ...
b4.txt	2176	ภัยพิบัติที่ร้ายแรงที่...
b5.txt	2198	การชื่นชมในความดี...
b6.txt	2278	----ข่าวในประเทศ---...
b7.txt	2505	ความเสมอภาคอัน...
b8.txt	2798	เชิญ 3 ทีมดังบอล...
b9.txt	2798	นิยามของความรัก...
b10.txt	2911	สัตว์สวย ป่างาม...
c1.txt	3088	ทะเบียนรักออนไลน์...
c2.txt	3228	ค่าไฟขึ้น 12.16...
c3.txt	3358	จดทะเบียน'วาเลน...

ตารางที่ 4.1 (ต่อ)

ชื่อแฟ้มข้อมูล	ปริมาณข้อมูล(หน่วยเป็นไบต์)	รายละเอียด
c4.txt	3369	บทที่ 1...
c5.txt	3499	คำทำนายประจำ...
c6.txt	3505	สัญญาณบอกเหตุ...
c7.txt	3675	ชาติไทยไม่สน...
c8.txt	3819	นร. ตาบอดฮือ...
c9.txt	3924	เตือนวาเลนไทน์...
c10.txt	3991	19 ข้อคิดในเรื่อง...
d1.txt	4146	10 ข้อคิด ชีวิต...
d2.txt	4304	บันได 7 ขั้นสู่ความ...
d3.txt	4436	ผู้ชายไม่ชอบ...
d4.txt	4471	บุกปล้นตามใบสั่ง...
d5.txt	4500	แบบบ้านเพื่อประชา...
d6.txt	4659	หิ้งหนังสือทักษิณ...
d7.txt	4663	คอลัมน์เก็บตกข่าว...
d8.txt	4760	วาไรตี้ : ม่วนซื่น...
d9.txt	4863	นายกฯรับปากดับไฟ...
d10.txt	5102	อกหัก...ไม่ยักตาย...
e1.txt	5155	ฝาดเฝื่อนแต่สุขใจ...
e2.txt	5222	บทที่ 1...
e3.txt	5449	จากบุฟเฟ่ต์คาบิเน็ต...
e4.txt	5598	บิ๊กชาติไทยเต้น...
e5.txt	5686	ทำไมพอเป็น...
e6.txt	5792	ช็อตเด็ดการตลาด...
e7.txt	5841	โบนัสมนุษย์เงิน...
e8.txt	5868	ชกเข้าเป้า...
e9.txt	5992	ห้องใต้หลังคา...
e10.txt	6112	วิธีดำรงความเป็น...
f1.txt	6190	ศักดิ์ดา โอดถูกเบรก...
f2.txt	6272	เข้มวาเลนไทน์...

ตารางที่ 4.1 (ต่อ)

ชื่อแฟ้มข้อมูล	ปริมาณข้อมูล(หน่วยเป็นไบต์)	รายละเอียด
f3.txt	6294	Boss และ Super...
f4.txt	6318	บุกปล้นบ้าน...
f5.txt	6422	ดูงานตำรวจ...
f6.txt	6767	พบติดหวัดนกอีก...
f7.txt	6793	วิกฤติหวัดนก...
f8.txt	6967	เต่าเสือดาว จีโอชี...
f9.txt	7114	สธ.รับหวัดนก...
f10.txt	7137	ศปร.3แฉอดีตบิ๊ก...
g1.txt	7244	ตลาดสาหร่ายโต...
g2.txt	7308	"สลากออมสินรี...
g3.txt	7327	ไชย ไชยวรรณ...
g4.txt	7438	อย่าแต่งงานเพราะ...
g5.txt	7527	ยุคแห่งความยุ่งยาก...
g6.txt	7591	ซูซูกิ แกรนด์...
g7.txt	7610	ความรัก 12 แบบ...
g8.txt	7626	ศุลกากรเดือดสูญเสีย...
g9.txt	7779	ตลาดแอร์สุขภาพ...
g10.txt	7889	คุมผับทำเหล้านอก...
h1.txt	8458	กระชากหน้ากาก...
h2.txt	8538	10 ล้านศพ...
h3.txt	8611	'พัชรี'ปลอม30บริษัท...
h4.txt	8628	จองคิวเชือด"บุหรี่...
h5.txt	8764	ISPหนาว-ทศท...
h6.txt	8783	มัจจุราชไข้หวัดนก...
h7.txt	8850	หนุนสร้างสนามบิน...
h8.txt	8853	ยิง-ฟัน 5 ราย...
h9.txt	8896	เปิดประตูค้าเสรี...
h10.txt	9105	ตังกุย สมุนไพร...
i1.txt	9256	อุ้มลูกหนี้ผ่านแผน...

ตารางที่ 4.1 (ต่อ)

ชื่อแฟ้มข้อมูล	ปริมาณข้อมูล(หน่วยเป็นไบต์)	รายละเอียด
i2.txt	9298	อกหักไม่ยักตาย (3)...
i3.txt	9355	อกหัก... ไม่ยักตาย (2)...
i4.txt	9367	อกหัก ไม่ยักตาย (6)...
i5.txt	9571	อกหัก แต่ไม่ยักตาย ...
i6.txt	9651	โรงเรียนชายแดน...
i7.txt	9905	อกหัก... ไม่ยักตาย 4...
i8.txt	9931	เรื่องของคนที่มีอาชีพ...
i9.txt	10096	หวั่นธปท.คุมค่าบาท...
i10.txt	10226	"ฮับน้ำมัน"เอเชีย...
j1.txt	19402	โทร. ครั้งที่ 1st...
j2.txt	29526	The Return of Media...
j3.txt	38942	เซ็นทรัล พาร์ค...
j4.txt	51759	ตร.กาม4นร.ไม่รอด...
j5.txt	61500	บางกอกโพสต์วุ่น...
j6.txt	70928	"อัยการ"สั่งฟ้อง...
j7.txt	80332	ล่าหัวโจกอุเทน...
j8.txt	90541	"ภูเก็ต"ฮือล้อม...
j9.txt	100267	นิโคลน้ำตาซึม...
j10.txt	200892	จับแก๊งโจ๋กาญจน์...

4.2 โปรแกรมบีบอัดข้อมูล

โปรแกรมที่ใช้ในการทดสอบการบีบอัดข้อมูล มีดังนี้

โปรแกรมที่พัฒนาจากอัลกอริทึมฮัฟฟ์แมน *

โปรแกรมที่พัฒนาจากอัลกอริทึมของวิธีการเชิงคำนวณ *

โปรแกรม PKZIP เวอร์ชัน 2.04 (ที่มา : http://www.frostburg.edu 2002)

โปรแกรม ARJ เวอร์ชัน 2.82 (ที่มา : http://www.arjsoft.com 2002)

โปรแกรม BZIP2 เวอร์ชัน 1.02 (ที่มา : http://www.andrewsworld.org 2002)

* ที่มา : Source code จากหนังสือ The Data Compression Book

4.3 เครื่องมือที่ใช้ในการวิจัย

เครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบ มีคุณสมบัติดังนี้

หน่วยประมวลผลกลาง(CPU)	:	AMD Duron 700 MHz
หน่วยความจำหลัก(RAM)	:	128 MB SD-RAM
หน่วยความจำสำรอง(Hard Disk)	:	20 GB
ระบบปฏิบัติการ(OS)	:	Windows XP Professional Version 2002
โปรแกรมที่ใช้ในการพัฒนา	:	Turbo C++ Version 3.0
โปรแกรมที่ใช้วิเคราะห์ข้อมูลทางสถิติ	:	Microsoft Excel Version 97
		SPSS Version 12.0 for Windows

4.4 การทดสอบ

ผู้วิจัยได้เก็บรวบรวมข้อมูลเพื่อใช้ในทำการทดสอบ ที่ระดับความเชื่อมั่น 95 เปอร์เซ็นต์ โดยมีวัตถุประสงค์ เพื่อเปรียบเทียบปริมาณข้อมูลระหว่างข้อมูลที่ผ่านการเข้ารหัสด้วยโปรแกรมการบีบอัดข้อมูลเพียงอย่างเดียว กับข้อมูลที่ผ่านทั้งการเข้ารหัสด้วยโปรแกรมการแปลงข้อความภาษาไทยและโปรแกรมการบีบอัดข้อมูล ซึ่งมีขั้นตอนการทดสอบ 3 ประการหลักดังนี้

ขั้นตอนแรก ผู้วิจัยจะนำตัวอย่างทั้ง 100 ตัวอย่าง มาผ่านการบีบอัดข้อมูลด้วยโปรแกรมการบีบอัดข้อมูลที่เตรียมไว้ (หัวข้อที่ 4.4.1)

ขั้นที่สอง ผู้วิจัยจะนำตัวอย่างทั้ง 100 ตัวอย่างมาทำการแปลงข้อมูลด้วยโปรแกรมการแปลงข้อความภาษาไทยทั้ง 3 วิธี (หัวข้อที่ 4.4.2 และหัวข้อที่ 4.4.3)

ขั้นที่สาม นำผลลัพธ์ที่ได้ในขั้นที่สองมาทำการบีบอัดข้อมูลด้วยโปรแกรมการบีบอัดข้อมูลที่เตรียมไว้ (หัวข้อที่ 4.4.4)

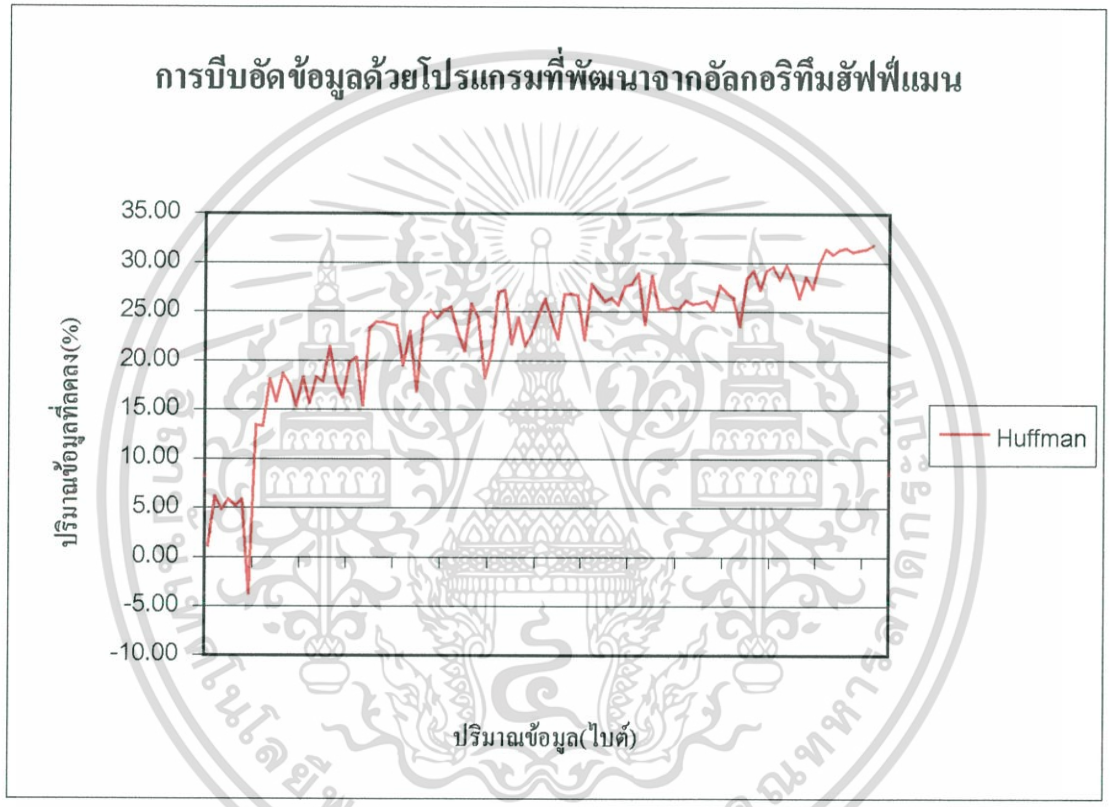
โดยรายละเอียดต่างๆ จะกล่าวถึงในลำดับต่อๆไป

4.4.1 การบีบอัดข้อมูล

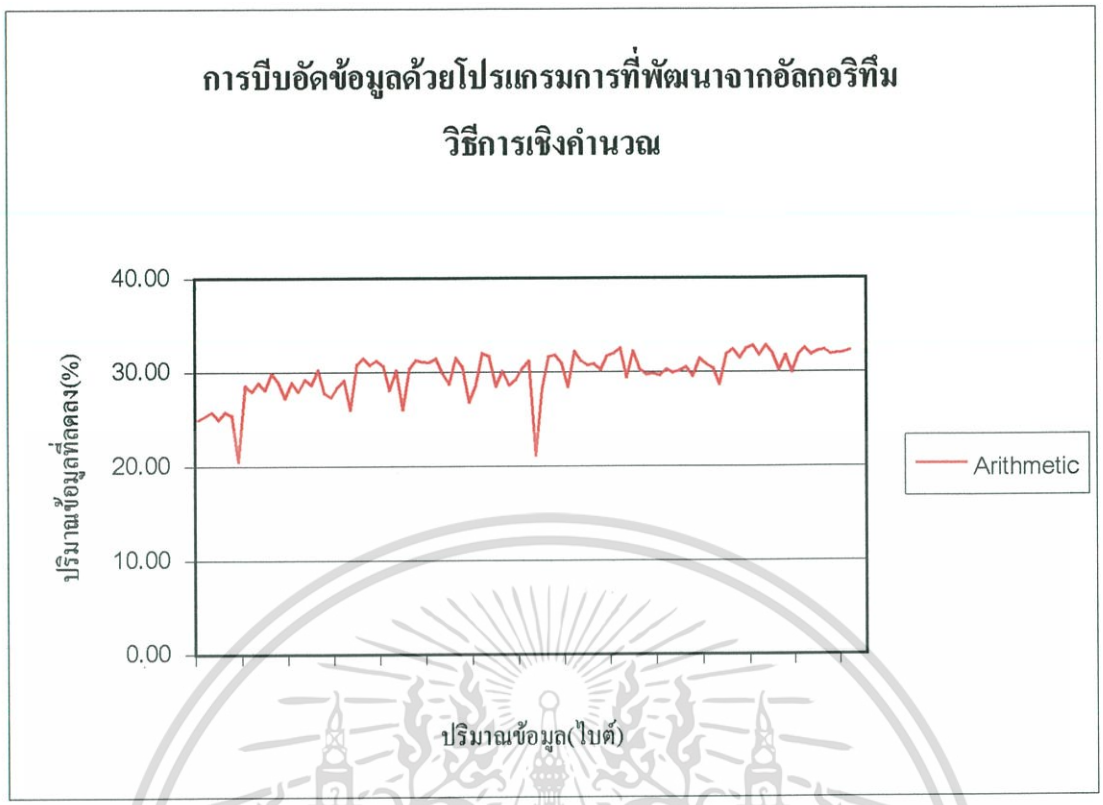
ทำการบีบอัดข้อมูลที่ได้จากการสุ่ม 100 ตัวอย่าง เพื่อหาปริมาณข้อมูลที่ลดลง ด้วยโปรแกรมการที่พัฒนาจากอัลกอริทึมฮัฟฟ์แมน โปรแกรมการที่พัฒนาจากอัลกอริทึมวิธีการเชิงคำนวณ โปรแกรม PKZIP, โปรแกรม ARJ และโปรแกรม BZIP2 ซึ่งผลลัพธ์ที่ได้แสดงในรูปแบบกราฟเส้น ในรูปที่ 4.1 รูปที่ 4.2 รูปที่ 4.3 รูปที่ 4.4 และรูปที่ 4.5 ตามลำดับ

เมื่อ แกนนอน คือ ปริมาณข้อมูลต้นกำเนิดที่เพิ่มขึ้น มีหน่วยเป็น ไบต์
(ขนาดของแฟ้มข้อมูล a1.txt ถึง j10.txt)
แกนตั้ง คือ ปริมาณข้อมูลที่ลดลงหลังผ่านการบีบอัดข้อมูล มีหน่วยเป็น เปอร์เซ็นต์

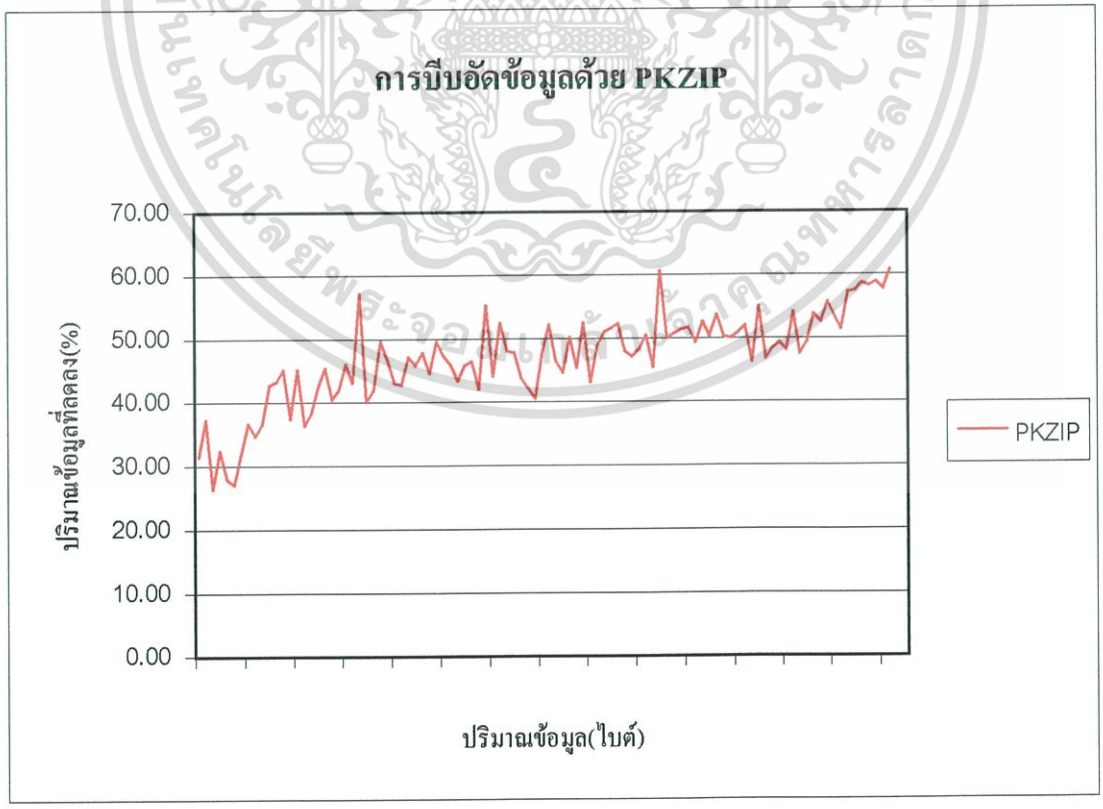
$$\text{ปริมาณข้อมูลที่ลดลง}(\%) = \frac{\text{ปริมาณข้อมูลต้นกำเนิด} - \text{ปริมาณข้อมูลหลังผ่านการบีบอัด}}{\text{ปริมาณข้อมูลต้นกำเนิด}} \text{x100} \qquad (4.1)$$



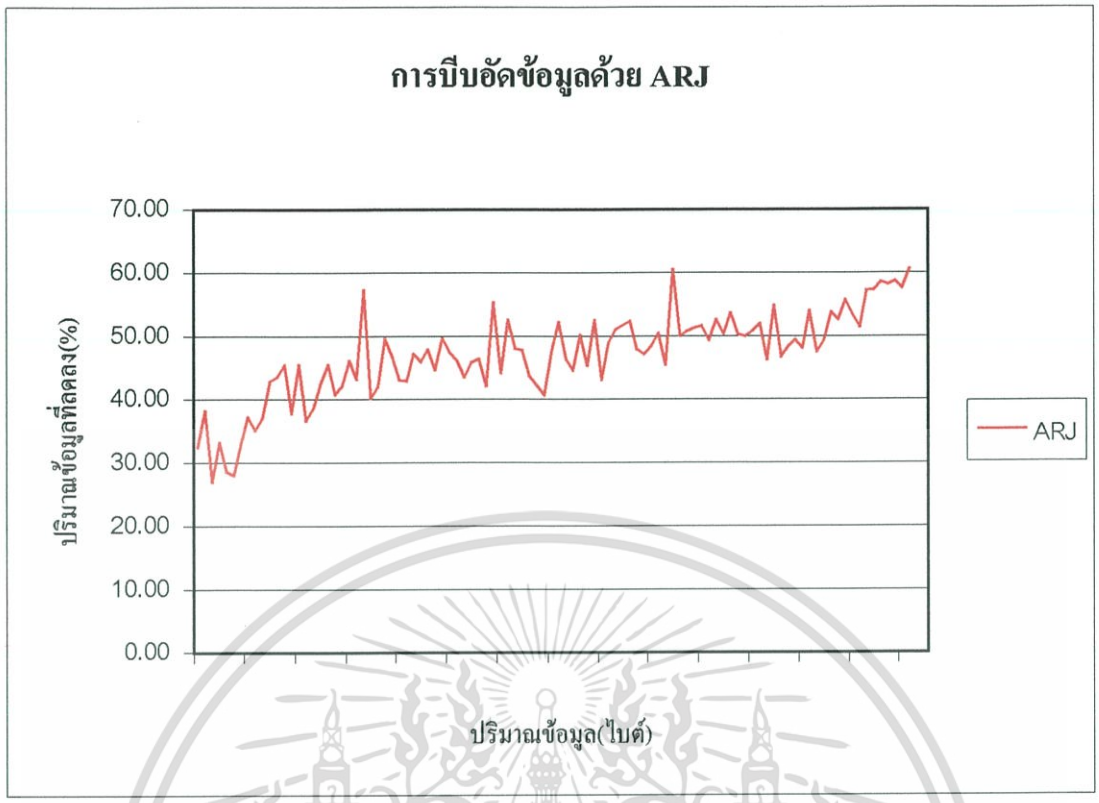
รูปที่ 4.1 กราฟแสดงปริมาณข้อมูลที่ลดลงหลังผ่านการบีบอัดข้อมูลด้วยโปรแกรมที่พัฒนาจากอัลกอริทึมฮัฟฟ์แมน



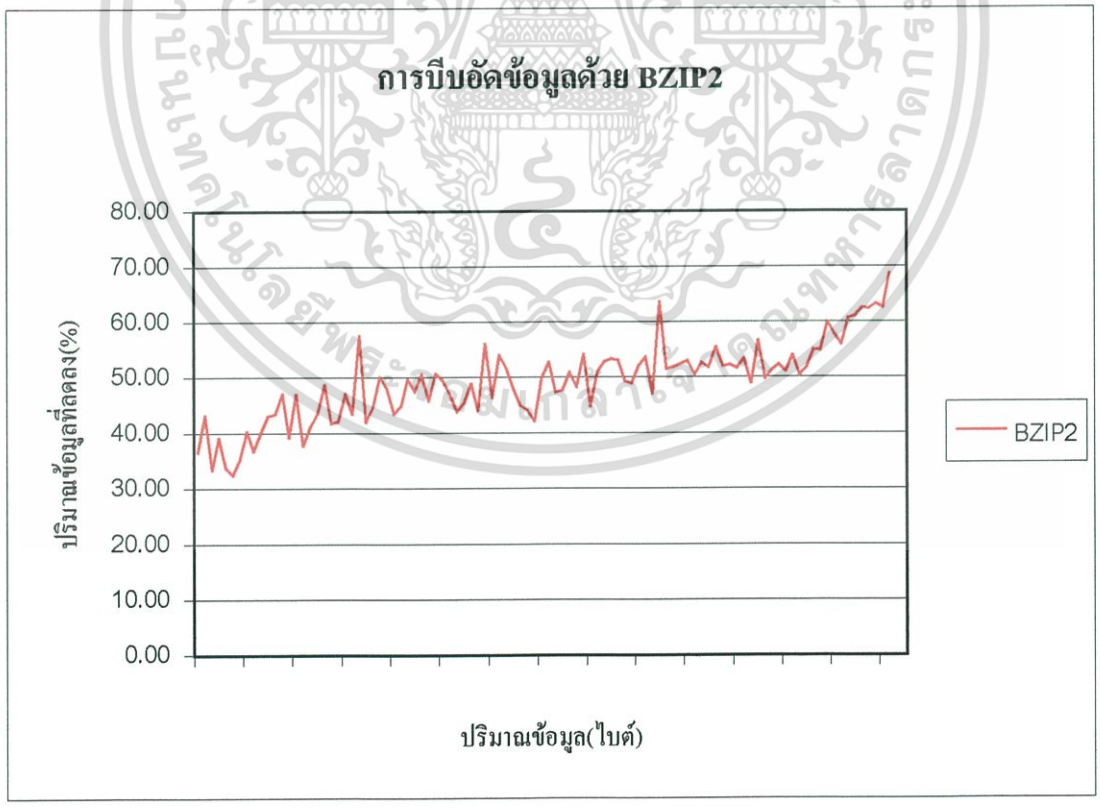
รูปที่ 4.2 กราฟแสดงปริมาณข้อมูลที่ลดลงหลังผ่านการบีบอัดข้อมูลด้วยโปรแกรมการที่พัฒนาจากอัลกอริทึมวิธีการเชิงคำนวณ



รูปที่ 4.3 กราฟแสดงปริมาณข้อมูลที่ลดลงหลังผ่านการบีบอัดข้อมูลด้วยโปรแกรม PKZIP



รูปที่ 4.4 กราฟแสดงปริมาณข้อมูลที่ลดลงหลังผ่านการบีบอัดข้อมูลด้วยโปรแกรม ARJ



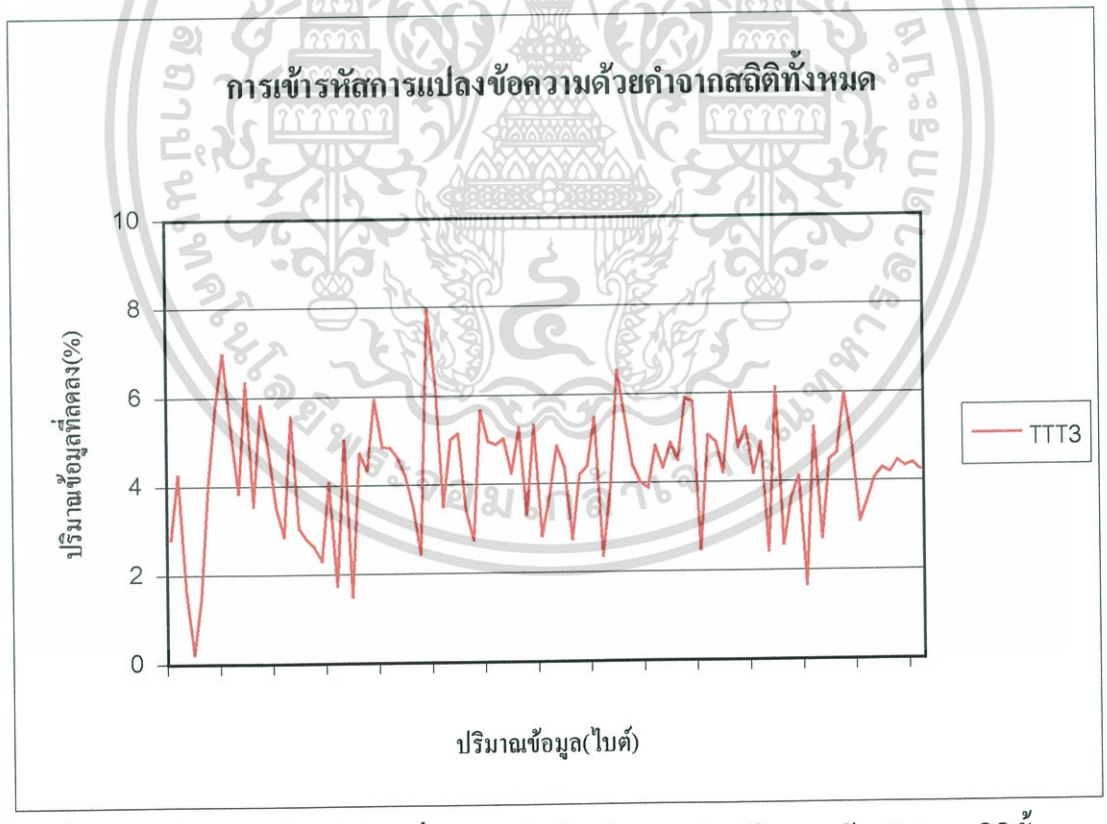
รูปที่ 4.5 กราฟแสดงปริมาณข้อมูลที่ลดลงหลังผ่านการบีบอัดข้อมูลด้วยโปรแกรม BZIP2

4.4.2 การเข้ารหัสการแปลงข้อมูล

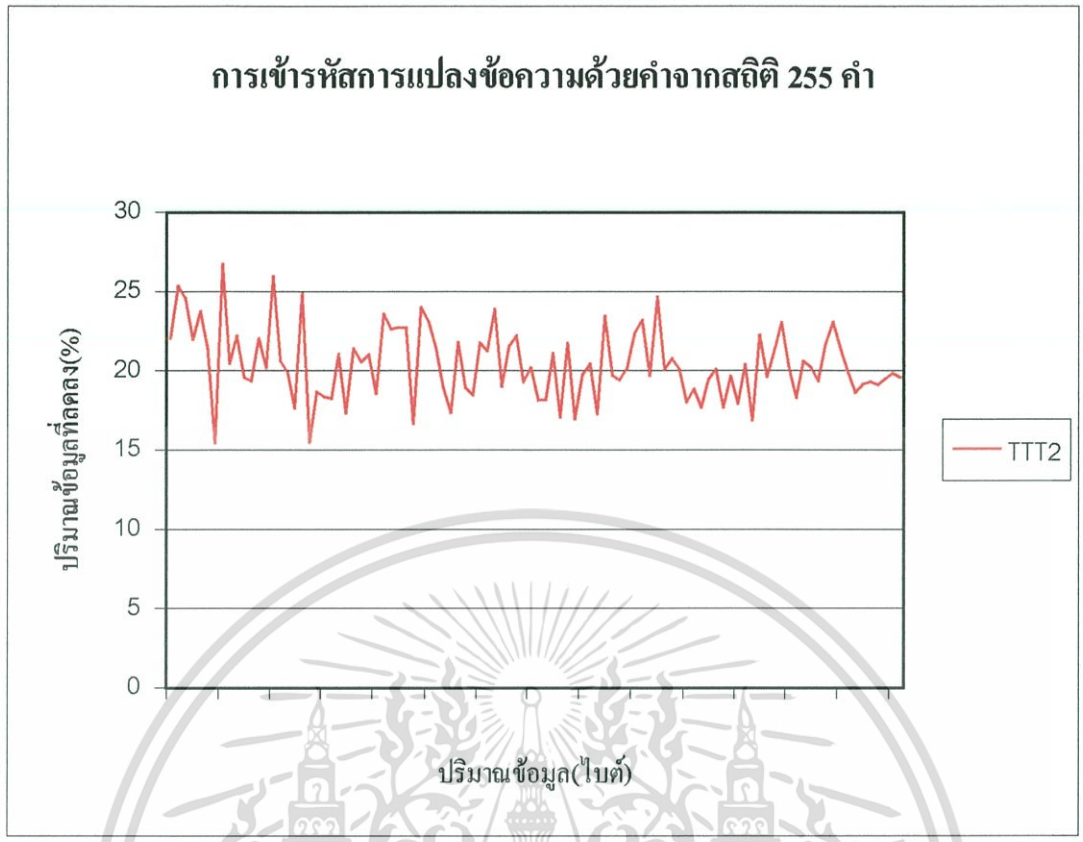
นำเอาข้อมูลที่เตรียมไว้ทั้ง 100 ตัวอย่าง มาผ่านการเข้ารหัสด้วย TTT3, TTT2 และ TTT1 ซึ่งผลที่ได้รับหลังเข้ารหัส ข้อมูลจะมีขนาดลดลง ดังแสดงในรูปที่ 4.6 และ รูปที่ 4.7 และ 4.8 ตามลำดับ สำหรับกราฟแสดงการเปรียบเทียบปริมาณข้อมูลหลังผ่านการเข้ารหัสด้วย TTT3, TTT2 และ TTT1 แสดงในรูปที่ 4.9

เมื่อ แกนนอน คือ ปริมาณข้อมูลต้นกำเนิดที่เพิ่มขึ้น มีหน่วยเป็น ไบต์(a1.txt ถึง j10.txt)
แกนตั้ง คือ ปริมาณข้อมูลที่ลดลงหลังผ่านการแปลงข้อมูล มีหน่วยเป็น เปอร์เซ็นต์

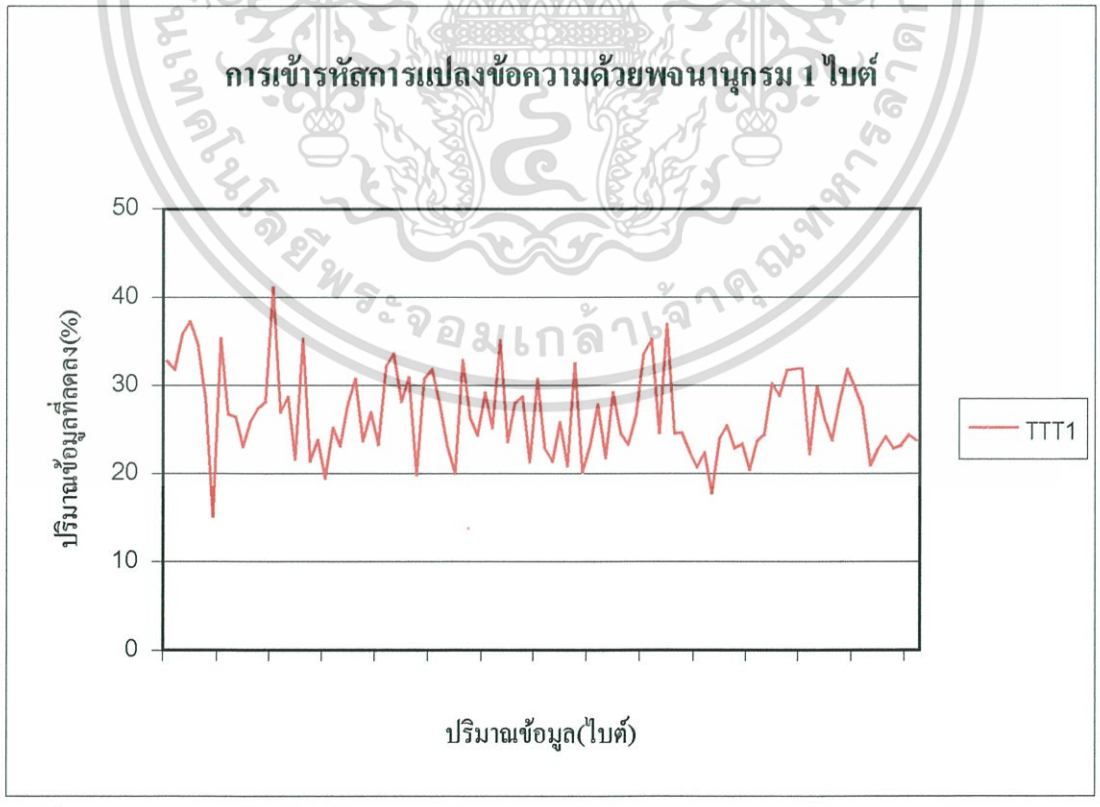
$$\text{ปริมาณข้อมูลที่ลดลง(\%)} = \frac{\text{ปริมาณข้อมูลต้นกำเนิด - ปริมาณข้อมูลหลังผ่านการแปลง}}{\text{ปริมาณข้อมูลต้นกำเนิด}} \text{x100} \quad (4.2)$$



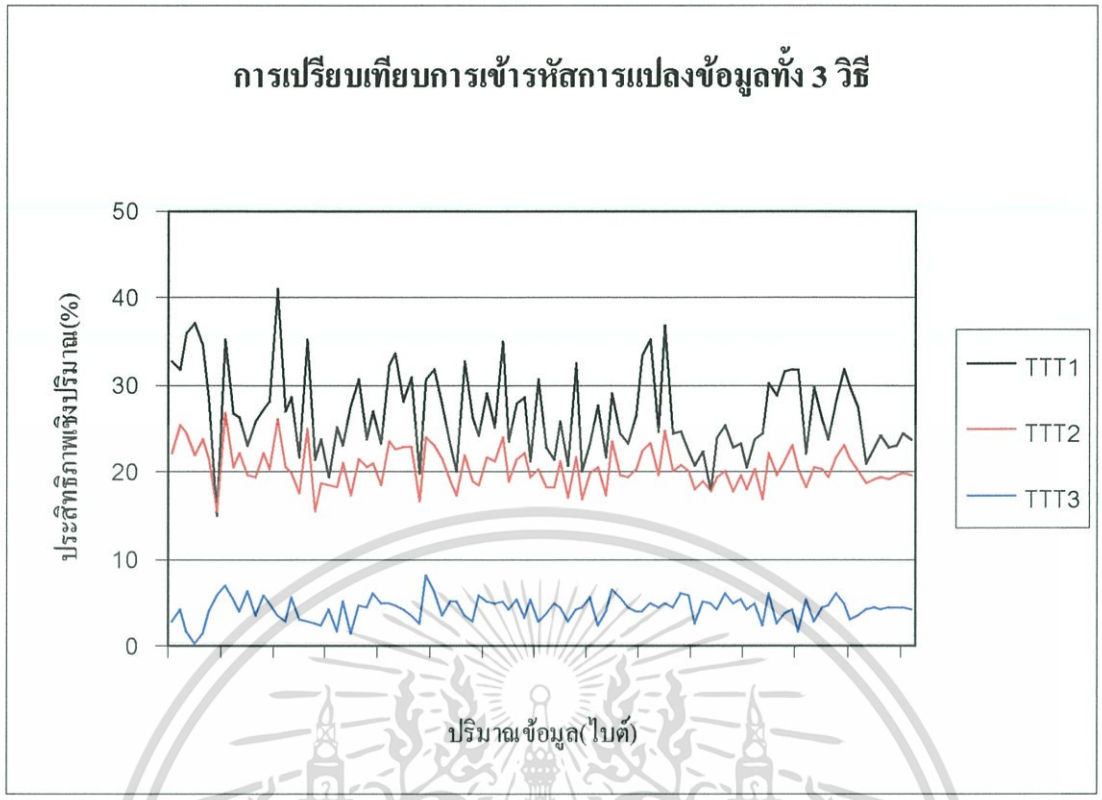
รูปที่ 4.6 กราฟแสดงปริมาณข้อมูลที่ลดลงหลังเข้ารหัสการแปลงข้อความด้วยคำจากสถิติทั้งหมด



รูปที่ 4.7 กราฟแสดงปริมาณข้อมูลที่ลดลงหลังเข้ารหัสการแปลงข้อความด้วยคำจากสถิติ 255 คำ



รูปที่ 4.8 กราฟแสดงปริมาณข้อมูลที่ลดลงหลังเข้ารหัสการแปลงข้อความด้วยพจนานุกรม 1 ไบต์



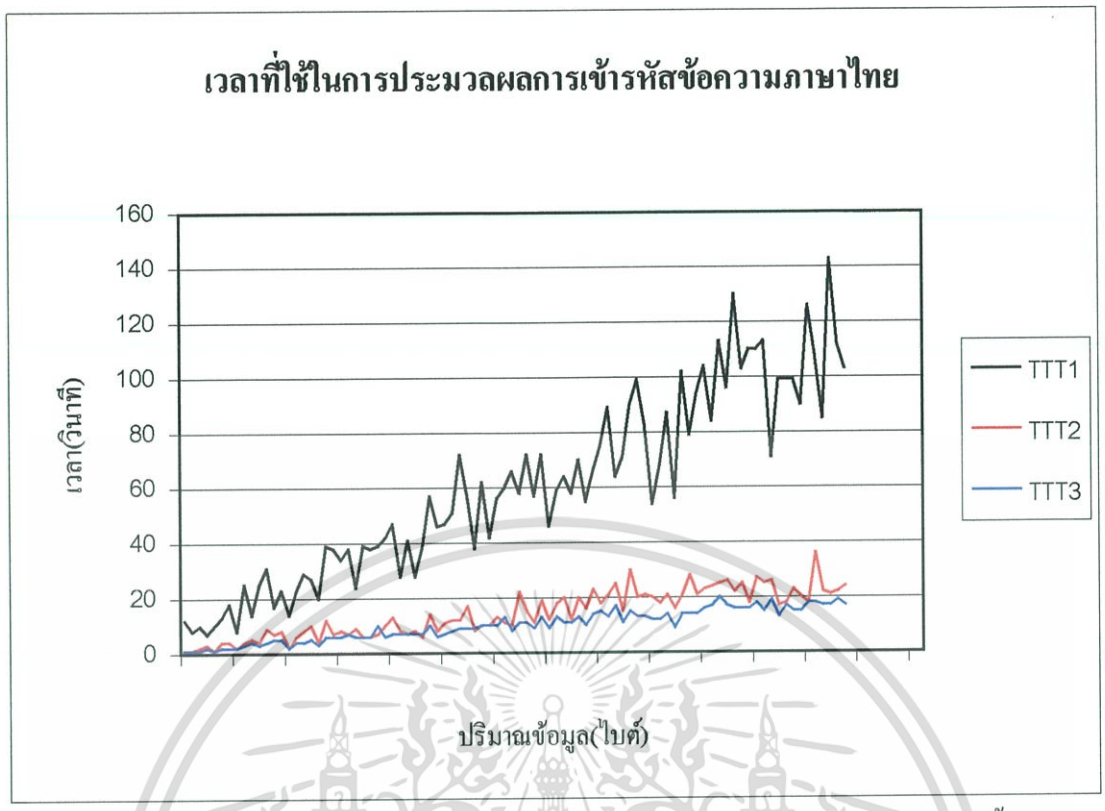
รูปที่ 4.9 กราฟแสดงการเปรียบเทียบปริมาณข้อมูลหลังผ่านการเข้ารหัสด้วย TTT3, TTT2 และ TTT1

4.4.3 เวลาที่ใช้ในเข้ารหัสและถอดรหัสการแปลงข้อมูล

เวลาที่ใช้ในการประมวลผลการเข้ารหัสการแปลงข้อความด้วยคำจากสถิติทั้งหมด วิธีการแปลงข้อความด้วยคำจากสถิติ 255 คำ และวิธีการแปลงข้อความด้วยพจนานุกรม 1 ไบต์ แสดงดังรูปที่ 4.10

เมื่อ แกนนอน คือ ปริมาณข้อมูลต้นกำเนิดที่เพิ่มขึ้น มีหน่วยเป็น ไบต์
แกนตั้ง คือ เวลาที่ใช้ในการประมวลผล มีหน่วยเป็น วินาที

สำหรับเวลาที่ใช้ในการประมวลผลการถอดรหัสการแปลงข้อความด้วยคำจากสถิติทั้งหมด วิธีการแปลงข้อความด้วยคำจากสถิติ 255 คำ และวิธีการแปลงข้อความด้วยพจนานุกรม 1 ไบต์ จะใช้เวลาประมาณ 1 วินาที



รูปที่ 4.10 กราฟแสดงเวลาที่ใช้ในการประมวลผลการเข้ารหัสข้อความภาษาไทยของทั้ง 3 วิธี

4.4.4 การบีบอัดข้อมูลที่เพิ่มส่วนการแปลงข้อมูล

นำแฟ้มข้อมูลที่ผ่านการเข้ารหัสการแปลงข้อมูลในหัวข้อที่ 4.4.2 ทั้งหมด มาทำการลดขนาดข้อมูลด้วยโปรแกรมการที่พัฒนาจากอัลกอริทึมมูลฮัฟฟ์แมน โปรแกรมการที่พัฒนาจากอัลกอริทึมวิธีการเชิงคำนวณ โปรแกรม PKZIP, โปรแกรม ARJ และโปรแกรม BZIP2 เพื่อนำผลที่ได้มาเปรียบเทียบประสิทธิภาพเชิงปริมาณ กับการลดขนาดด้วยวิธีการบีบอัดข้อมูลเพียงขั้นตอนเดียว ซึ่งให้ผลการทดสอบแสดงในรูปแบบกราฟเส้นดังนี้

กราฟแสดงการเปรียบเทียบระหว่าง การลดขนาดข้อมูลด้วยโปรแกรมที่พัฒนาจากอัลกอริทึมฮัฟฟ์แมนเพียงอย่างเดียว กับการลดขนาดข้อมูลด้วยโปรแกรมการที่พัฒนาจากอัลกอริทึมฮัฟฟ์แมนที่เพิ่มส่วนการแปลงข้อความภาษาไทย TTT3, TTT2 และ TTT1 แสดงดังรูปที่ 4.11

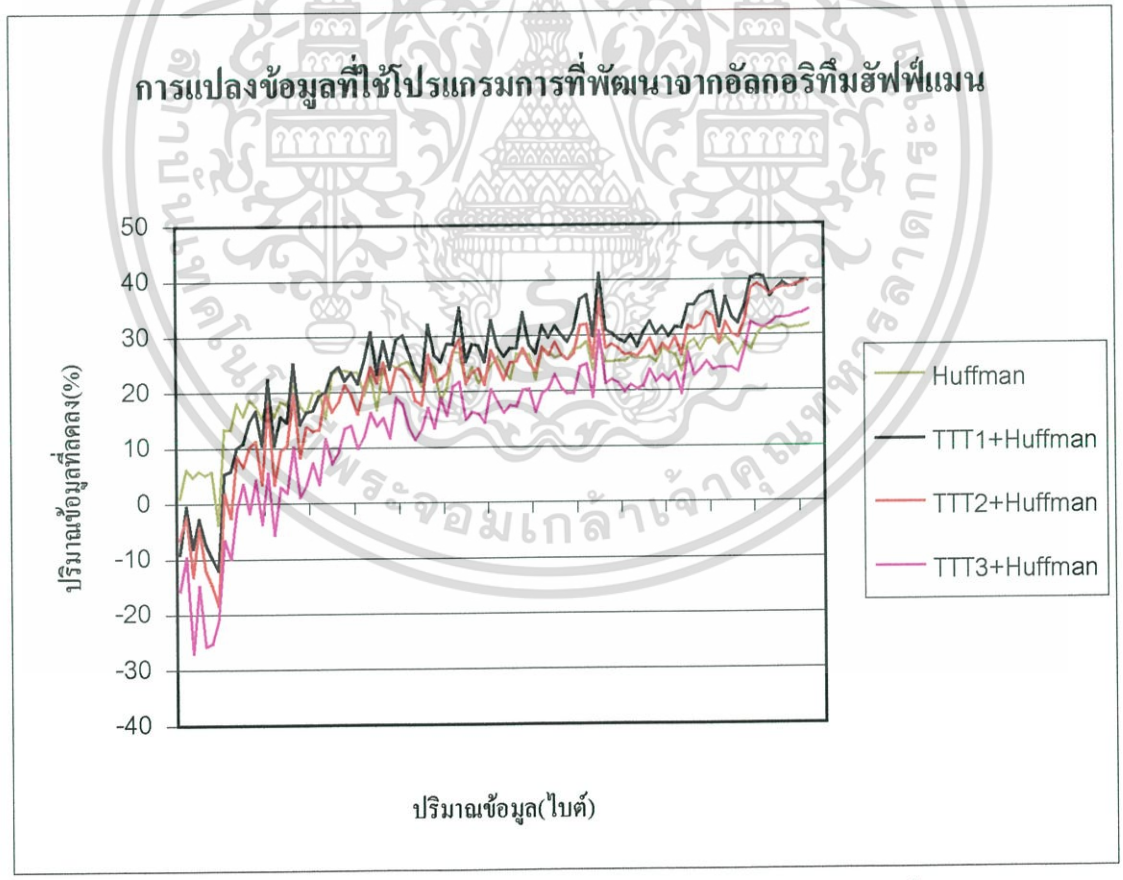
กราฟแสดงการเปรียบเทียบระหว่าง การลดขนาดข้อมูลด้วยโปรแกรมที่พัฒนาจากอัลกอริทึมวิธีการเชิงคำนวณเพียงอย่างเดียว กับการลดขนาดข้อมูลด้วยโปรแกรมที่พัฒนาจากอัลกอริทึมวิธีการเชิงคำนวณที่เพิ่มส่วนการแปลงข้อความภาษาไทย TTT3, TTT2 และ TTT1 แสดงดังรูปที่ 4.12

กราฟแสดงการเปรียบเทียบระหว่าง การลดขนาดข้อมูลด้วยโปรแกรม PKZIP เพียงอย่างเดียว กับการลดขนาดข้อมูลด้วยโปรแกรม PKZIP ที่เพิ่มส่วนการแปลงข้อความภาษาไทย TTT3, TTT2 และ TTT1 แสดงดังรูปที่ 4.13

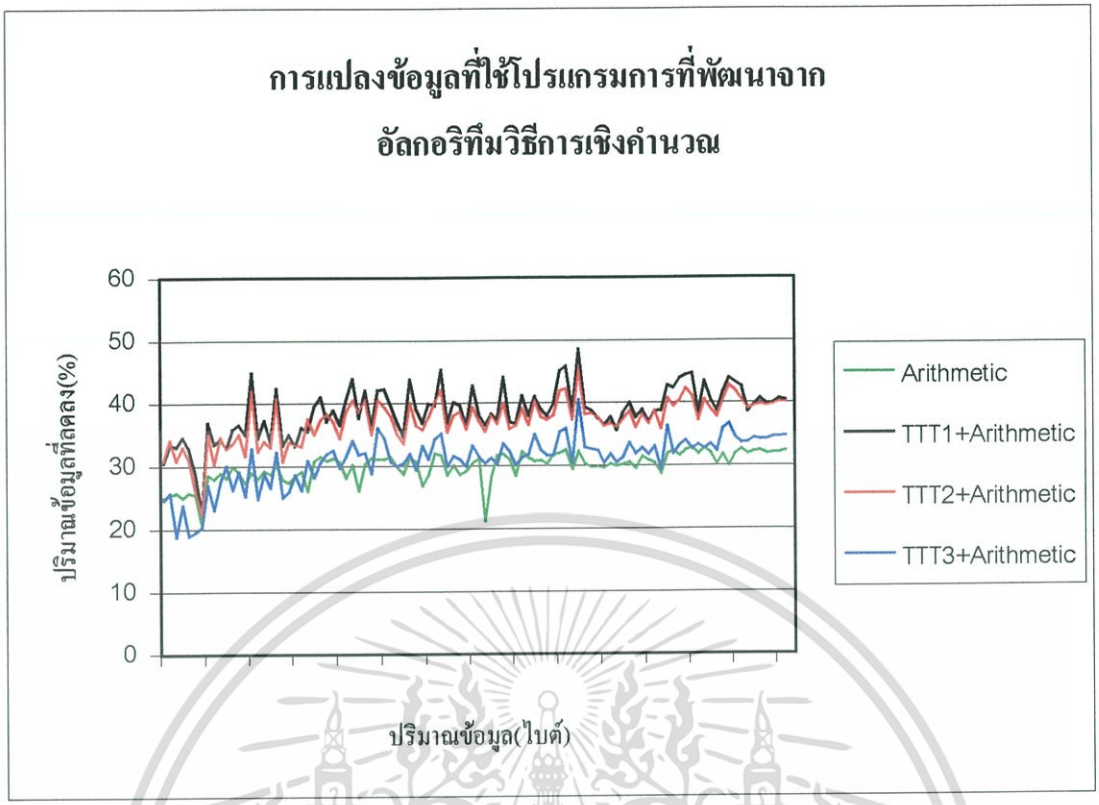
กราฟแสดงการเปรียบเทียบระหว่าง การลดขนาดข้อมูลด้วยโปรแกรม ARJ เพียงอย่างเดียว กับการลดขนาดข้อมูลด้วยโปรแกรม ARJ ที่เพิ่มส่วนการแปลงข้อความภาษาไทย TTT3, TTT2 และ TTT1 แสดงดังรูปที่ 4.14

กราฟแสดงการเปรียบเทียบระหว่าง การลดขนาดข้อมูลด้วยโปรแกรม BZIP2 เพียงอย่างเดียว กับการลดขนาดข้อมูลด้วยโปรแกรม BZIP2 ที่เพิ่มส่วนการแปลงข้อความภาษาไทย TTT3, TTT2 และ TTT1 แสดงดังรูปที่ 4.15

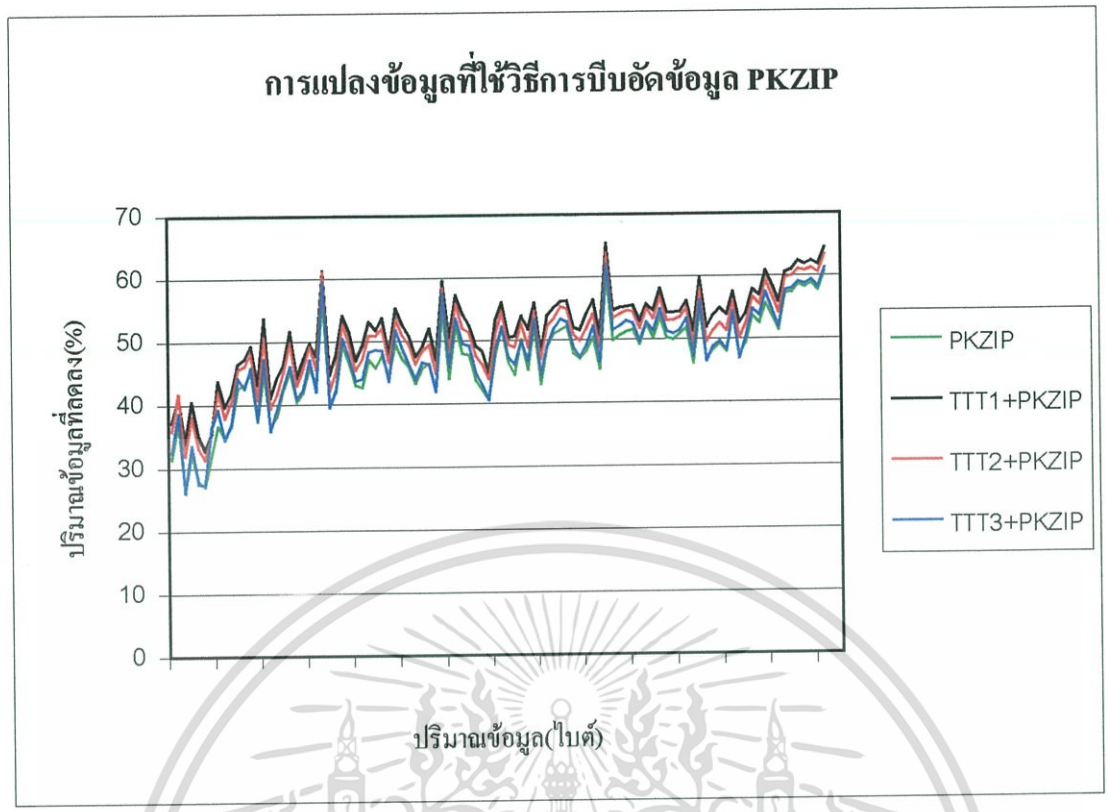
เมื่อ แกนนอน คือ ปริมาณข้อมูลต้นกำเนิดที่เพิ่มขึ้น มีหน่วยเป็น ไบต์
(ขนาดของข้อมูล a1.txt ถึง j10.txt)
แกนตั้ง คือ ปริมาณข้อมูลหลังผ่านการบีบอัดข้อมูลที่เพิ่มขึ้น มีหน่วยเป็น เปอร์เซ็นต์



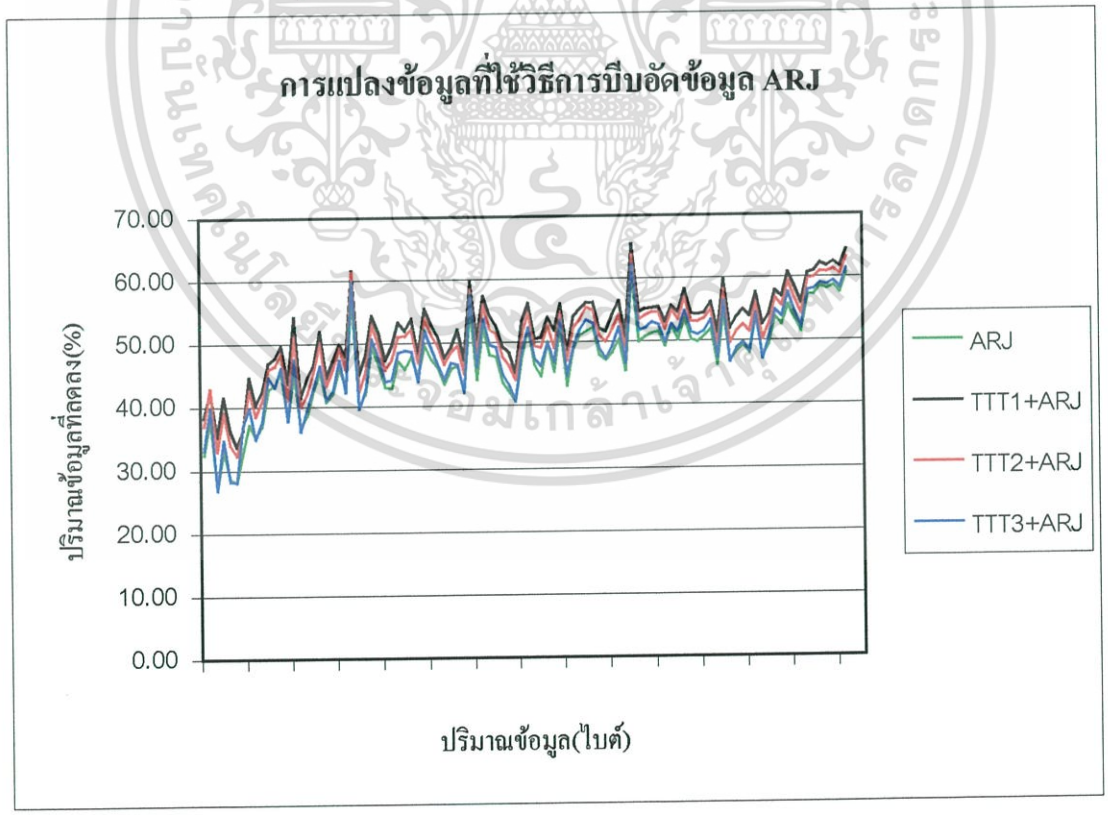
รูปที่ 4.11 กราฟแสดงการเปรียบเทียบระหว่าง การลดขนาดข้อมูลด้วยโปรแกรมที่พัฒนาจากอัลกอริทึมฮัฟฟ์แมนเพียงอย่างเดียว กับการลดขนาดข้อมูลด้วยโปรแกรมการที่พัฒนาจากอัลกอริทึมฮัฟฟ์แมนที่เพิ่มส่วนการแปลงข้อความภาษาไทย TTT3, TTT2 และ TTT1



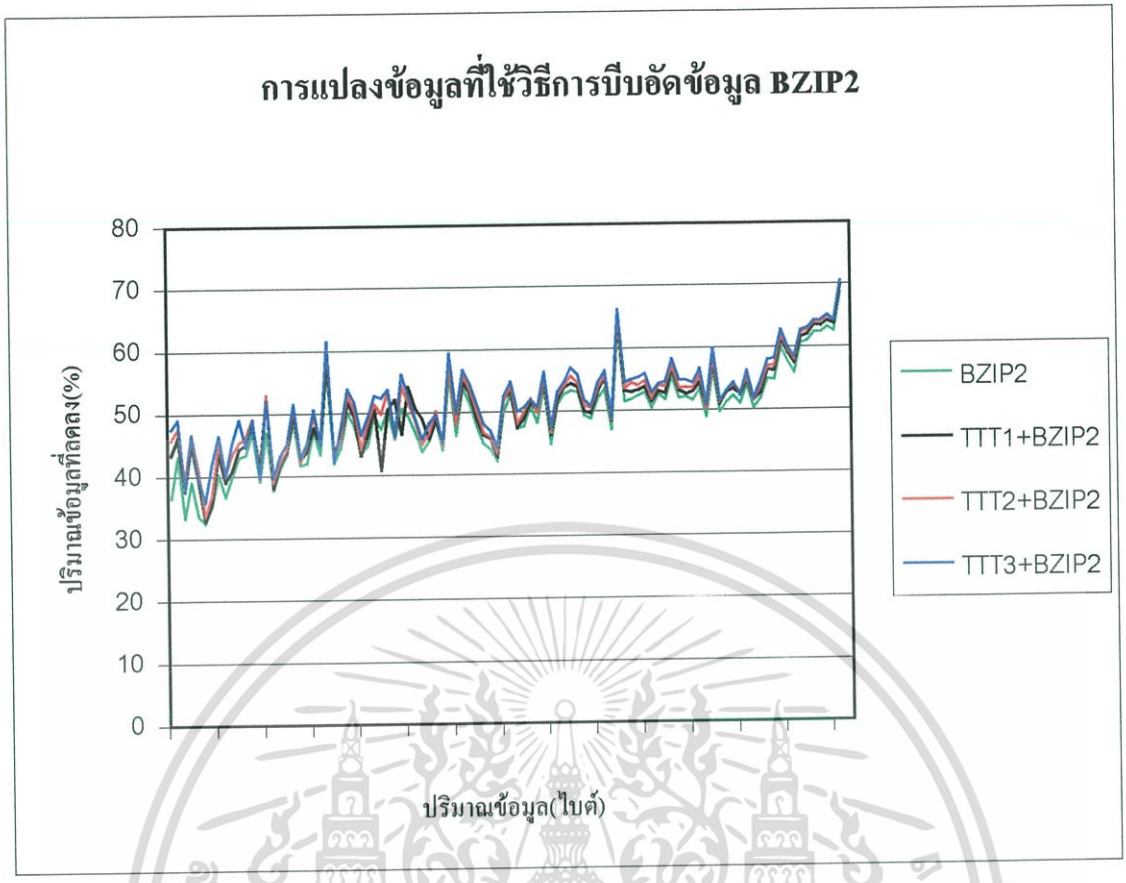
รูปที่ 4.12 กราฟแสดงการเปรียบเทียบระหว่าง การลดขนาดข้อมูลด้วยโปรแกรมที่พัฒนาจากอัลกอริทึมวิธีการเชิงคำนวณเพียงอย่างเดียว กับการลดขนาดข้อมูลด้วยโปรแกรมที่พัฒนาจากอัลกอริทึมวิธีการเชิงคำนวณที่เพิ่มส่วนการแปลงข้อความภาษาไทย TTT3, TTT2 และ TTT1



รูปที่ 4.13 กราฟแสดงการเปรียบเทียบระหว่าง การลดขนาดข้อมูลด้วยโปรแกรม PKZIP กับ การลดขนาดข้อมูลด้วยโปรแกรม PKZIP ที่เพิ่มส่วน TTT3, TTT2 และ TTT1



รูปที่ 4.14 กราฟแสดงการเปรียบเทียบระหว่าง การลดขนาดข้อมูลด้วยโปรแกรม ARJ กับ การลดขนาดข้อมูลด้วยโปรแกรม ARJ ที่เพิ่มส่วน TTT3, TTT2 และ TTT1



รูปที่ 4.15 กราฟแสดงการเปรียบเทียบระหว่าง การลดขนาดข้อมูลด้วยโปรแกรม BZIP2 กับการลดขนาดข้อมูลด้วยโปรแกรม BZIP2 ที่เพิ่มส่วน TTT3, TTT2 และ TTT1

4.5 การหาค่าเฉลี่ยปริมาณข้อมูลที่ลดลง

ค่าเฉลี่ยปริมาณข้อมูลที่ลดลง คือ ค่าเฉลี่ยถ่วงน้ำหนักของปริมาณข้อมูลที่ลดลง ซึ่งสามารถหาได้จากสมการดังต่อไปนี้

$$\text{ค่าเฉลี่ยปริมาณข้อมูลที่ลดลง}(\%) = \frac{\text{ผลรวมปริมาณข้อมูลที่ลดลงทั้งหมด}}{100} \qquad (4.3)$$

4.5.1 ค่าเฉลี่ยปริมาณข้อมูลที่ลงลดหลังเข้ารหัสการแปลงข้อมูล

เมื่อเข้ารหัสการแปลงข้อความด้วยคำจากสถิติ ข้อมูลจะมีขนาดลดลง 4.2646 เปอร์เซ็นต์

เมื่อเข้ารหัสการแปลงข้อความด้วยคำจากสถิติ 255 คำ ข้อมูลจะมีขนาดลดลง 19.8324 เปอร์เซ็นต์

เมื่อเข้ารหัสวิธีการแปลงข้อความด้วยพจนานุกรม 1 ไบต์ ข้อมูลจะมีขนาดลดลง 24.9525 เปอร์เซ็นต์

4.5.2 ค่าเฉลี่ยปริมาณข้อมูลที่ลดลงหลังบีบอัดข้อมูลเพียงอย่างเดียว

เมื่อบีบอัดข้อมูลด้วยโปรแกรมการที่พัฒนาจากอัลกอริทึมฮัฟฟ์แมน ข้อมูลจะมีขนาดลดลง 28.6452 เปอร์เซ็นต์

เมื่อบีบอัดข้อมูลด้วยโปรแกรมการที่พัฒนาจากอัลกอริทึมวิธีการเชิงคำนวณ ข้อมูลจะมีขนาดลดลง 31.3343 เปอร์เซ็นต์

เมื่อบีบอัดข้อมูลด้วยโปรแกรม PKZIP ข้อมูลจะมีขนาดลดลง 54.1294 เปอร์เซ็นต์

เมื่อบีบอัดข้อมูลด้วยโปรแกรม ARJ ข้อมูลจะมีขนาดลดลง 54.0691 เปอร์เซ็นต์

เมื่อบีบอัดข้อมูลด้วยโปรแกรม BZIP2 ข้อมูลจะมีขนาดลดลง 57.9305 เปอร์เซ็นต์

4.5.3 ค่าเฉลี่ยปริมาณข้อมูลที่ลงลดหลังบีบอัดข้อมูลและเข้ารหัสด้วย TTT3

เมื่อเข้ารหัสด้วย TTT3 และบีบอัดข้อมูลด้วยโปรแกรมการที่พัฒนาจากอัลกอริทึมฮัฟฟ์-แมน ข้อมูลจะมีขนาดลดลง 27.2346 เปอร์เซ็นต์

เมื่อเข้ารหัสด้วย TTT3 และบีบอัดข้อมูลด้วยโปรแกรมการที่พัฒนาจากอัลกอริทึมวิธีการเชิงคำนวณ ข้อมูลจะมีขนาดลดลง 33.3682 เปอร์เซ็นต์

เมื่อเข้ารหัสด้วย TTT3 และบีบอัดข้อมูลด้วยโปรแกรม PKZIP ข้อมูลจะมีขนาดลดลง 54.8231 เปอร์เซ็นต์

เมื่อเข้ารหัสด้วย TTT3 และบีบอัดข้อมูลด้วยโปรแกรม ARJ ข้อมูลจะมีขนาดลดลง 54.905 เปอร์เซ็นต์

เมื่อเข้ารหัสด้วย TTT3 และบีบอัดข้อมูลด้วยโปรแกรม BZIP2 ข้อมูลจะมีขนาดลดลง 60.1681 เปอร์เซ็นต์

4.5.4 ค่าเฉลี่ยปริมาณข้อมูลที่ลดลงหลังบีบอัดข้อมูลและเข้ารหัสด้วย TTT2

เมื่อเข้ารหัสด้วย TTT2 และบีบอัดข้อมูลด้วยโปรแกรมการที่พัฒนาจากอัลกอริทึมฮัฟฟ์-แมน ข้อมูลจะมีขนาดลดลง 33.3672 เปอร์เซ็นต์

เมื่อเข้ารหัสด้วย TTT2 และบีบอัดข้อมูลด้วยโปรแกรมการที่พัฒนาจากอัลกอริทึมวิธีการเชิงคำนวณ ข้อมูลจะมีขนาดลดลง 39.123 เปอร์เซ็นต์

เมื่อเข้ารหัสด้วย TTT2 และบีบอัดข้อมูลด้วยโปรแกรม PKZIP ข้อมูลจะมีขนาดลดลง 57.0019 เปอร์เซ็นต์

เมื่อเข้ารหัสด้วย TTT2 และบีบอัดข้อมูลด้วยโปรแกรม ARJ ข้อมูลจะมีขนาดลดลง 57.0008 เปอร์เซ็นต์

เมื่อเข้ารหัสด้วย TTT2 และบีบอัดข้อมูลด้วยโปรแกรม BZIP2 ข้อมูลจะมีขนาดลดลง 59.705 เปอร์เซ็นต์

4.5.5 ค่าเฉลี่ยปริมาณข้อมูลที่ลดลงหลังบีบอัดข้อมูลและเข้ารหัสด้วย TTT1

เมื่อเข้ารหัสด้วย TTT1 และบีบอัดข้อมูลด้วยโปรแกรมการที่พัฒนาจากอัลกอริทึมฮัฟฟ์-แมน ข้อมูลจะมีขนาดลดลง 35.0974 เปอร์เซ็นต์

เมื่อเข้ารหัสด้วย TTT1 และบีบอัดข้อมูลด้วยโปรแกรมการที่พัฒนาจากอัลกอริทึมวิธีการเชิงคำนวณ ข้อมูลจะมีขนาดลดลง 40.0621 เปอร์เซ็นต์

เมื่อเข้ารหัสด้วย TTT1 และบีบอัดข้อมูลด้วยโปรแกรม PKZIP ข้อมูลจะมีขนาดลดลง 58.389 เปอร์เซ็นต์

เมื่อเข้ารหัสด้วย TTT1 และบีบอัดข้อมูลด้วยโปรแกรม ARJ ข้อมูลจะมีขนาดลดลง 58.3892 เปอร์เซ็นต์

เมื่อเข้ารหัสด้วย TTT1 และบีบอัดข้อมูลด้วยโปรแกรม BZIP2 ข้อมูลจะมีขนาดลดลง 59.1513 เปอร์เซ็นต์

4.6 การวิเคราะห์ข้อมูล

งานวิจัยนี้ได้นำข้อมูลมาวิเคราะห์ด้วยโปรแกรม SPSS เวอร์ชัน 12.0 โดยใช้การทดสอบค่าทีของความแตกต่างระหว่างค่า 2 ค่าเฉลี่ยที่ไม่เป็นอิสระต่อกันแบบจับคู่(paired t-test) [11]

4.6.1 การทดสอบค่าทีแบบจับคู่

ทำการวิเคราะห์เปรียบเทียบค่าเฉลี่ยปริมาณข้อมูลที่ลดลง ด้วยการทดสอบค่าทีของความแตกต่างระหว่างค่า 2 ค่าเฉลี่ยที่ไม่เป็นอิสระต่อกันแบบจับคู่ ระหว่างการบีบอัดข้อมูลเพียงอย่างเดียว(B) กับการบีบอัดข้อมูลที่เพิ่มส่วนการแปลงข้อความภาษาไทย TTT3, TTT2 และ TTT1(A) โดยมีข้อตกลงเบื้องต้น ดังต่อไปนี้

1) การทดสอบค่า t ที่ระดับนัยสำคัญ 0.05
2) ข้อมูลแต่ละกลุ่มไม่เป็นอิสระต่อกัน
3) ค่า $t_{ตาราง} = 1.662$(ค่า $t_{ตาราง}$ ได้จากการเทียบบัญญัติไตรยางค์จากตาราง t มาตรฐาน เมื่อองศาแห่งความเป็นอิสระ = 99)
4) ให้ B_i และ A_i เป็น ปริมาณข้อมูลของข้อมูลคู่ที่ i เมื่อ $i = 1,2,..., 100$

สมมติฐาน

$$H_0 \quad : \quad \mu_d \le d_0$$
$$H_1 \quad : \quad \mu_d > d_0 \quad , \quad d_0 = 0$$

สถิติทดสอบ

$$t_{คำนวณ} = \frac{\bar{d} - d_0}{S_d / \sqrt{n}} \tag{4.4}$$

$$\bar{d} = \frac{\sum_{i=1}^{n} d_i}{n} \text{ เมื่อ } d_i = B_i - A_i \text{ และ } i=1,2,...,n \tag{4.5}$$

$$S_d = \sqrt{\frac{\sum_{i=1}^{n}(di - \bar{d})^2}{(n-1)}} \text{ หรือ } \sqrt{\frac{\sum_{i=1}^{n} d_i^{\,2} - \frac{(\sum_{i=1}^{n} d_i)^2}{n}}{(n-1)}} \tag{4.6}$$

เมื่อ $t_{คำนวณ}$ แทน ค่าพิจารณาใน t-distribution

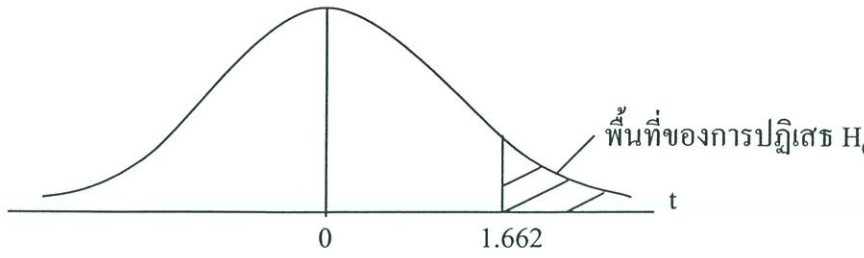
d_i แทน ค่าความแตกต่างระหว่างปริมาณข้อมูลคู่ที่ i

n แทน จำนวนคู่ข้อมูลที่ใช้ในการทดลอง

S_d แทน ส่วนเบี่ยงเบนมาตรฐานของค่าความแตกต่างระหว่างข้อมูลคู่ที่ i

การตัดสินใจ

จะปฏิเสธ H_0 ถ้าค่า $t_{คำนวณ}$ ที่ได้มีค่ามากกว่าค่า $t_{ตาราง}$ ($t_{คำนวณ}$<1.662) นั่นคือตกในบริเวณวิกฤต โดยพื้นที่ของการปฏิเสธ H_0 ดังนี้



กรณีที่ค่า $t_{คำนวณ} < t_{ตาราง}$ หมายถึง ยอมรับว่าค่าเฉลี่ยปริมาณข้อมูลที่ผ่านการบีบอัดข้อมูลน้อยกว่าหรือเท่ากับค่าเฉลี่ยปริมาณข้อมูลที่ผ่านการบีบอัดข้อมูลและเพิ่มส่วนการเข้ารหัสข้อความภาษาไทย อย่างมีนัยสำคัญที่ 0.05

กรณีที่ค่า $t_{คำนวณ} > t_{ตาราง}$ หมายถึง ยอมรับว่าค่าเฉลี่ยปริมาณข้อมูลที่ผ่านการบีบอัดข้อมูลมากกว่าค่าเฉลี่ยปริมาณข้อมูลที่ผ่านการบีบอัดข้อมูลและเพิ่มส่วนการเข้ารหัสข้อความภาษาไทย อย่างมีนัยสำคัญที่ 0.05

4.7 ผลการวิเคราะห์ข้อมูล

ผลการวิเคราะห์ข้อมูลระหว่างการเข้ารหัสด้วยโปรแกรมการบีบอัดข้อมูลเพียงอย่างเดียวกับการเข้ารหัสข้อมูลที่ใช้ทั้งโปรแกรมการแปลงข้อความภาษาไทยและโปรแกรมการบีบอัด แสดงดังตารางที่ 4.2

ตารางที่ 4.2 ผลการทดสอบค่าทีแบบจับคู่ ที่ระดับนัยสำคัญ 0.05 เมื่อค่า $t_{ตาราง} = 1.662$

การบีบอัดข้อมูล	การแปลงข้อมูล	$\overline{X}_B$	$\overline{X}_A$	$t_{คำนวณ}$	H_0	H_1
วิธีฮัฟฟ์แมน	TTT3	8874.91	9050.36	-2.337	ยอมรับ	ปฏิเสธ
	TTT2		8287.61	2.756	ปฏิเสธ	ยอมรับ
	TTT1		8072.41	3.8	ปฏิเสธ	ยอมรับ
วิธีการเชิงคำนวณ	TTT3	8540.45	8287.48	3.96	ปฏิเสธ	ยอมรับ
	TTT2		7571.72	4.746	ปฏิเสธ	ยอมรับ
	TTT1		8287.48	5.086	ปฏิเสธ	ยอมรับ
PKZIP	TTT3	5705.26	5175.46	5.966	ปฏิเสธ	ยอมรับ
	TTT2		5347.99	5.311	ปฏิเสธ	ยอมรับ
	TTT1		5618.98	5.394	ปฏิเสธ	ยอมรับ

ตารางที่ 4.2 (ต่อ)

การบีบอัดข้อมูล	การแปลงข้อมูล	$\overline{x}_B$	$\overline{x}_A$	$t_{คำนวณ}$	H_0	H_1
ARJ	TTT3	5712.76	5175.44	5.423	ปฏิเสธ	ยอมรับ
	TTT2		5348.13	5.389	ปฏิเสธ	ยอมรับ
	TTT1		5608.76	5.426	ปฏิเสธ	ยอมรับ
BZIP2	TTT3	5232.49	5080.65	5.759	ปฏิเสธ	ยอมรับ
	TTT2		5011.78	4.870	ปฏิเสธ	ยอมรับ
	TTT1		4954.18	4.822	ปฏิเสธ	ยอมรับ

ผลการวิเคราะห์การแปลงข้อมูล พบว่ามีกรณีเดียวที่ค่า $t_{คำนวณ}$ น้อยกว่า $t_{ตาราง}$ ซึ่งเป็นกรณีที่ทำการทดสอบประสิทธิภาพระหว่างการลดขนาดข้อมูลด้วยโปรแกรมการบีบอัดข้อมูลฮัฟฟ์แมนเพียงอย่างเดียว กับการลดขนาดข้อมูลด้วยโปรแกรมการบีบอัดข้อมูลฮัฟฟ์แมนและโปรแกรมการการแปลงข้อความด้วยคำจากสถิติทั้งหมด ซึ่งสรุปผลการวิเคราะห์ข้อมูลได้ว่าค่าเฉลี่ยปริมาณข้อมูลที่ผ่านการบีบอัดข้อมูลด้วยวิธีฮัฟฟ์แมนน้อยกว่าหรือเท่ากับค่าเฉลี่ยปริมาณข้อมูลที่เพิ่มส่วนการเข้ารหัสข้อความภาษาไทย อย่างมีนัยสำคัญที่ 0.05

สำหรับปริมาณข้อมูลที่ลดลงของวิธีการแปลงข้อความภาษาไทยทั้ง 3 วิธีสามารถสรุปได้โดยแสดงในตารางที่ 4.3 และตารางที่ 4.4

ตารางที่ 4.3 สรุปปริมาณข้อมูลที่ลดลงเมื่อเทียบกับข้อมูลต้นกำเนิด มีหน่วยเป็น เปอร์เซ็นต์

การแปลงข้อมูล / การบีบอัดข้อมูล	ไม่มี	TTT3	TTT2	TTT1
ไม่มี	-	4.2646	19.8324	24.9525
วิธีฮัฟฟ์แมน	28.6452	27.2346	33.3672	35.0974
วิธีการเชิงคำนวณ	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

ตารางที่ 4.4 ประสิทธิภาพที่เพิ่มขึ้นของการลดขนาดข้อมูล มีหน่วยเป็น เปอร์เซ็นต์

การแปลงข้อมูล / การบีบอัดข้อมูล	TTT3	TTT2	TTT1
วิธีฮัฟฟ์แมน	-1.4106	4.7220	6.4522
วิธีการเชิงคำนวณ	2.0339	7.7887	8.7278
PKZIP	0.6937	2.8725	4.2596
ARJ	0.8359	2.9317	4.3201
BZIP2	2.2376	1.7745	1.2208

จากตารางที่ 4.3 และ 4.4 สามารถสรุปวิธีการแปลงข้อความที่เหมาะสมที่สุดสำหรับปริมาณการลดขนาดของข้อมูลสูงสุดของแต่ละ โปรแกรมการบีบอัดข้อมูล ที่ระดับนัยสำคัญที่ 0.05 ได้ดังต่อไปนี้

TTT1 ช่วยให้การลดขนาดข้อมูลเพิ่มขึ้นมากที่สุด เมื่อใช้ร่วมกับโปรแกรมการบีบอัดข้อมูลที่พัฒนาจากอัลกอริทึมฮัฟฟ์แมน โปรแกรมการบีบอัดข้อมูลที่พัฒนาจากอัลกอริทึมวิธีการเชิงคำนวณ โปรแกรมการบีบอัดข้อมูล PKZIP หรือโปรแกรมการบีบอัดข้อมูล ARJ

TTT3 ช่วยให้การลดขนาดข้อมูลเพิ่มขึ้นมากที่สุด เมื่อใช้ร่วมกับโปรแกรมการบีบอัดข้อมูล BZIP2

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยเรื่องการแปลงข้อความภาษาไทยเพื่อการบีบอัดข้อมูลนี้ มีวัตถุประสงค์เพื่อเพิ่มประสิทธิภาพการลดขนาดข้อมูลประเภทข้อความภาษาไทย โดยข้อมูลที่ผ่านการเข้ารหัสด้วยวิธีการแปลงข้อความก่อนนำไปบีบอัดข้อมูลนี้ จะมีขนาดลดลงกว่าข้อมูลที่ผ่านขั้นตอนการบีบอัดข้อมูลเพียงอย่างเดียว ซึ่งหลักการที่นำมาใช้ในการแปลงข้อความภาษาไทยนี้ คือ ทำให้ข้อมูลเกิดความซ้ำซ้อนกันมากขึ้น อีกทั้งข้อมูลจะต้องมีขนาดเล็กลงในระดับหนึ่ง โดยนำเอาคำไทยจากสถิติข้อมูลคำไทย [1, 2] (ในตารางที่ ก1 ในภาคผนวก ก) มาใช้เป็นคำศัพท์ในการเข้ารหัสและถอดรหัสการแปลงข้อมูล ซึ่งในขั้นแรกผู้ทำวิทยานิพนธ์ได้ใช้คำไทยจากสถิติทั้งหมดในการพัฒนา พบว่าทำให้ข้อมูลมีขนาดลดลงหลังผ่านการเข้ารหัสเล็กน้อย จึงดำเนินการทดลองต่อไป โดยใช้สมมติฐานว่าการใช้คำจากสถิติที่พบบ่อยในลำดับแรกๆ จะช่วยลดขนาดข้อมูลได้เช่นกัน จึงพัฒนาวิธีการแปลงข้อความภาษาไทยขึ้นอีก 2 วิธี โดยใช้คำไทยจากสถิติ 255 ลำดับแรก และ 109 ลำดับแรกตามลำดับ ซึ่งการแปลงข้อความภาษาไทยทั้ง 2 วิธีนี้ สามารถลดขนาดข้อมูลได้มากขึ้น แต่มีข้อเสียในเรื่องของความยืดหยุ่นในการเพิ่มเติมคำศัพท์ โดยผลลัพธ์ที่ได้จากการทดลองการแปลงข้อความภาษาไทยทั้ง 3 วิธี สามารถสรุปได้ดังต่อไปนี้

1) เมื่อเพิ่มส่วนการแปลงข้อความด้วยคำจากสถิติทั้งหมด ให้กับการบีบอัดข้อมูลด้วยโปรแกรม BZIP2 จะทำให้สามารถลดขนาดของข้อมูลได้เพิ่มขึ้นกว่าการบีบอัดข้อมูลเพียงอย่างเดียวมากที่สุด คือ ข้อมูลต้นกำเนิดจะมีขนาดลดลงประมาณ 60 เปอร์เซ็นต์(เพิ่มขึ้นจากเดิมประมาณ 2 เปอร์เซ็นต์) ที่ระดับนัยสำคัญ 0.05

2) การบีบอัดข้อมูลด้วยวิธีฮัฟฟ์แมนที่เพิ่มส่วนการเข้ารหัสการแปลงข้อความด้วยคำจากสถิติทั้งหมด มีประสิทธิภาพการลดขนาดข้อมูลน้อยกว่าการบีบอัดข้อมูลเพียงอย่างเดียว แต่แนวโน้มของอัตราการลดขนาดข้อมูลจะเพิ่มขึ้นเมื่อข้อมูลมีปริมาณมากๆ

3) อัตราการลดขนาดข้อมูลของการเข้ารหัสการแปลงข้อความภาษาไทยวิธีการแปลงข้อความด้วยคำจากสถิติทั้งหมด วิธีการแปลงข้อความด้วยคำจากสถิติ 255 คำ และวิธีการแปลงข้อความด้วยพจนานุกรม 1 ไบต์ ด้วยวิธีการบีบอัดข้อมูลด้วย PKZIP ARJ และ BZIP2 มีแนวโน้มเพิ่มขึ้นอย่างสม่ำเสมอ (รูปที่ 4.13, 4.14 และ 4.15)

4) การเข้ารหัสด้วยวิธีการแปลงข้อความด้วยคำจากสถิติทั้งหมดใช้เวลาในการประมวลผลน้อยที่สุด และการเข้ารหัสด้วยวิธีการแปลงข้อความด้วยพจนานุกรม 1 ไบต์ใช้เวลาในการประมวลผลมากที่สุด

5) การถอดรหัสของทั้ง 3 วิธี ใช้เวลาน้อยเมื่อเทียบกับการเข้ารหัส

6) วิธีการแปลงข้อความด้วยคำจากสถิติทั้งหมด และวิธีการแปลงข้อความด้วยคำจากสถิติ 255 คำ มีความซับซ้อนน้อยที่สุด อีกทั้งใช้เวลาในการเข้ารหัสน้อยที่สุด

7) พจนานุกรมการแปลงข้อความภาษาไทยทั้ง 3 วิธี มีขนาดเล็กและสามารถปรับเปลี่ยนแก้ไขคำศัพท์ได้ง่าย

8) ข้อมูลทางสถิติคำไทยเป็นส่วนสำคัญการแปลงข้อความภาษาไทย

9) ผลการแปลงข้อความทั้งหมด ขึ้นอยู่กับสถิติการวิเคราะห์ข้อมูลคำไทยในตารางที่ ก1 ในภาคผนวก ก [1, 2]

10) นอกจากการเข้ารหัสการแปลงข้อความภาษาไทยทั้ง 3 วิธี จะสามารถทำให้ข้อมูลมีขนาดลดลงเพิ่มมากขึ้นแล้ว ผลพลอยได้ของการเข้ารหัสการแปลงข้อความภาษาไทยอีกประการหนึ่ง คือ ข้อมูลจะถูกเก็บรักษาเป็นความลับ เนื่องจากการเข้ารหัสการแปลงข้อความภาษาไทยจะทำให้ข้อมูลต้นกำเนิดเปลี่ยนเป็นข้อมูลที่เป็นรหัสที่ไม่สามารถตีความได้หากไม่มีตัวถอดรหัส ซึ่งช่วยลดปัญหาการโจรกรรมข้อมูล

5.2 ข้อเสนอแนะ

1) วิธีการแปลงข้อความภาษาไทยทั้ง 3 วิธี สามารถนำไปประยุกต์ใช้กับระบบการติดต่อสื่อสาร เนื่องจากมีการอ่านข้อมูลทีละตัวอักษร ซึ่งสามารถแปลงข้อมูลในขณะที่ยังอ่านข้อมูลไม่หมดทั้งแฟ้มข้อมูลได้ทันที

2) หากจะนำไปประยุกต์ใช้เป็นโปรแกรมการลดขนาดข้อมูล ควรมีอัลกอริทึมที่ใช้ตัดสินว่าจะใช้การแปลงข้อมูลหรือไม่ เนื่องจากว่ามีบางกรณีที่การแปลงข้อมูลไม่ทำให้การลดขนาดของข้อมูลมีประสิทธิภาพมากขึ้น

3) หากต้องการนำวิธีการแปลงข้อความภาษาไทยด้วยคำจากสถิติทั้งหมดไปประยุกต์ใช้สามารถปรับเปลี่ยนคำศัพท์ที่ใช้บ่อยตามความต้องการได้

5.3 แนวทางในการวิจัยต่อไป

1) งานวิจัยนี้อาศัยสถิติข้อมูลคำไทยที่ได้ในปีพุทธศักราช 2527 ซึ่งข้อมูลสถิติคำไทยในปัจจุบันอาจมีการเปลี่ยนแปลง

2) การแปลงข้อมูลทั้ง 3 วิธีที่กล่าวมา สามารถใช้ได้กับข้อมูลคำศัพท์ที่เก็บไว้เท่านั้น ยังไม่สามารถเรียนรู้รูปแบบคำหรือวลีอื่นๆที่เกิดขึ้นนอกเหนือจากคำที่อยู่ในพจนานุกรมได้

3) งานวิจัยนี้มุ่งเน้นการแปลงข้อมูลประเภทข้อความภาษาไทยโดยใช้พจนานุกรมคำไทย ถ้ามีการออกแบบให้พจนานุกรมเก็บคำศัพท์ได้หลายภาษาก็จะทำให้สามารถลดขนาดข้อมูลของเอกสารทั่วๆไปได้

4) ควรจะมีการทดลองต่อไป เพื่อหาข้อสรุปให้ได้ว่าควรจะใช้คำจากข้อมูลทางสถิติร้อยละเท่าไร จึงจะเพียงพอต่อการลดขนาดข้อมูล ซึ่งงานวิจัยนี้จะช่วยสนับสนุนแนวทางการวิจัยในหัวข้อที่ 3

เอกสารอ้างอิง

[1] ยืน ภู่วรวรรณ. **"การวิเคราะห์ข้อมูลคำไทย."** ห้องปฏิบัติการวิจัยไมโครคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเกษตรศาสตร์, การประชุมวิชาการทางวิศวกรรมไฟฟ้า 8 สถาบันอุดมศึกษา ครั้งที่ 7 เล่ม 3 คอมพิวเตอร์, ภาควิชาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตธนบุรี, ธันวาคม 2527.

[2] ยืน ภู่วรวรรณ, วิวรรธน์ อิ่มอารมณ์. **"การแบ่งแยกพยางค์ไทยด้วยดิกชันนารี."** ห้องปฏิบัติการวิจัยไมโครคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเกษตรศาสตร์, การประชุมทางวิชาการวิศวกรรมไฟฟ้า สถาบันอุดมศึกษาแห่งประเทศไทย ครั้งที่ 9, หอประชุมศูนย์วิทยาศาสตร์สุขภาพ มหาวิทยาลัยขอนแก่น, ธันวาคม 2529.

[3] ยืน ภู่วรวรรณ, สมนึก คีรีโต. **"ข้อเสนอแนะเกี่ยวกับมาตรฐานรหัสภาษาไทย."** การประชุมทางวิชาการวิศวกรรมไฟฟ้า ครั้งที่ 6 เล่ม 2 คอมพิวเตอร์และวิศวกรรมสื่อสาร มหาวิทยาลัยสงขลานครินทร์-หาดใหญ่, พฤศจิกายน 2526.

[4] ยืน ภู่วรวรรณ, ชัยยงค์ วงศ์ชัยสุวัฒน์. **"การออกแบบและลดขนาดข้อมูลคำไทย ในพจนานุกรมสำหรับงาน พิสูจน์อักษร."** วิทยาสารเกษตรศาสตร์ สาขาวิทยาศาสตร์ ปีที่ 23 ฉบับที่ 4, ตุลาคม - ธันวาคม 2532.

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

[6] Arturo San Emeterio Campos. "BWT: A Transformation Algorithm." [Online]. Available : http://www.arturocampos.com/ac_bwt.html. 2001.

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

[8] Awan F. and Mukherjee A. **"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. 2001.

[9] Kruse H., Mukherjee A. **"Preprocessing Text to Improve Compression Ratios."** Proceedings of the IEEE Data Compression Conference 1998, Snowbird, p. 556

[10] Dissunrat K. **"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. 2001.

เอกสารอ้างอิง(ต่อ)

[11] กัลยา วานิชย์บัญชา. **สถิติเพื่อการตัดสินใจ.** กรุงเทพมหานคร : โรงพิมพ์จุฬาลงกรณ์มหาวิทยาลัย. 2538.

ภาคผนวก ก

ตารางที่ ก1 การแจกแจงความถี่ของคำที่ใช้ในชีวิตประจำวัน 511 คำ

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
1	ที่	3474	2.6009	2.6009
2	การ	2726	2.0409	4.6418
3	เป็น	2073	1.5520	6.1938
4	ได้	1936	1.4494	7.6432
5	จะ	1874	1.4030	9.0462
6	ใน	1766	1.3222	10.3684
7	มี	1727	1.2930	11.6614
8	ไม่	1568	1.1739	12.8353
9	ก็	1552	1.1619	13.9972
10	ของ	1541	1.1537	15.1509
11	ให้	1486	1.1125	16.2634
12	ว่า	1471	1.1013	17.3647
13	ไป	1442	1.0796	18.4443
14	และ	1423	1.0654	19.5097
15	มา	1244	0.9314	20.4411
16	ความ	1159	0.8677	21.3088
17	ประ	1079	0.8078	22.1166
18	นี้	1066	0.7981	22.9147
19	ทำ	868	0.6499	23.5646
20	คน	856	0.6409	24.2055
21	ผู้	844	0.6319	24.8374
22	กิน	827	0.6192	25.4566
23	แล้ว	827	0.6192	26.0758
24	แต่	827	0.6192	26.6950
25	จาก	788	0.5900	27.2850
26	อย่าง	775	0.5802	27.8652
27	นั้น	766	0.5735	28.4387
28	อยู่	742	0.5555	28.9942
29	กับ	691	0.5173	29.5115
30	ต้อง	662	0.4956	30.0071

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
31	ทาง	612	0.4582	30.4653
32	เรือ	561	0.4200	30.8853
33	งาน	549	0.4110	31.2963
34	ด้วย	549	0.4110	31.7073
35	ใจ	531	0.3976	32.1049
36	ขึ้น	497	0.3721	32.4770
37	ถึง	495	0.3706	32.8476
38	ต่อ	481	0.3601	33.2077
39	เข้า	475	0.3556	33.5633
40	รับ	462	0.3459	33.9092
41	น้ำ	460	0.3444	34.2536
42	ยัง	460	0.3444	34.5980
43	เรา	458	0.3429	34.9409
44	มาก	456	0.3414	35.2823
45	โดย	451	0.3377	35.6200
46	ทั้ง	442	0.3309	35.9509
47	หน้า	430	0.3220	36.2729
48	วัน	420	0.3144	36.5873
49	ซึ่ง	414	0.3100	36.8973
50	ออก	412	0.3085	37.2058
51	คุณ	410	0.3070	37.5128
52	ใช้	408	0.3055	37.8183
53	ดี	396	0.2965	38.1148
54	เขา	390	0.2920	38.4068
55	ตัว	383	0.2867	38.6935
56	ตาม	381	0.2852	38.9787
57	คือ	380	0.2845	39.2632
58	รู้	380	0.2845	39.5477
59	เพราะ	379	0.2838	39.8315
60	จำ	372	0.2785	40.1100

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
61	เมื่อ	368	0.2755	40.3855
62	ปี	366	0.2740	40.6595
63	เรื่อง	364	0.2725	40.9320
64	อีก	362	0.2710	41.2030
65	ไทย	359	0.2688	41.4718
66	บ้าน	340	0.2546	41.7264
67	พระ	327	0.2448	41.9712
68	เห็น	325	0.2433	42.2145
69	ทุก	313	0.2343	42.4488
70	ผม	305	0.2283	42.6771
71	จึง	303	0.2269	42.9040
72	มัน	302	0.2261	43.1301
73	เทศ	301	0.2254	43.3555
74	กระ	299	0.2239	43.5794
75	หนึ่ง	299	0.2239	43.8033
76	เสีย	297	0.2224	44.0257
77	รถ	296	0.2216	44.2473
78	เงิน	294	0.2201	44.4674
79	กว่า	292	0.2186	44.6860
80	ตั้ง	290	0.2171	44.9031
81	หา	288	0.2156	45.1187
82	เพื่อ	287	0.2149	45.3336
83	ท่าน	282	0.2111	45.5447
84	ลง	280	0.2096	45.7543
85	เมือง	277	0.2074	45.9617
86	นัก	267	0.1999	46.1616
87	นาย	259	0.1939	46.3555
88	หลาย	257	0.1924	46.5479
89	ไว้	255	0.1909	46.7388
90	เอา	254	0.1902	46.9290

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
91	เวลา	253	0.1894	47.1184
92	เกิด	244	0.1827	47.3011
93	ละ	242	0.1812	47.4823
94	ผล	239	0.1789	47.6612
95	จัด	238	0.1782	47.8394
96	ดู	236	0.1767	48.0161
97	ชน	230	0.1722	48.1883
98	ถ้า	229	0.1714	48.3597
99	จน	229	0.1714	48.5311
100	ส่วน	229	0.1714	48.7025
101	บาง	227	0.1699	48.8724
102	ใหญ่	227	0.1699	49.0423
103	เอง	221	0.1655	49.2078
104	นำ	220	0.1647	49.3725
105	เลย	216	0.1617	49.5342
106	ครั้ง	214	0.1602	49.6944
107	หนัง	213	0.1595	49.8539
108	ก่อน	212	0.1587	50.0126
109	เดิน	206	0.1542	50.1668
110	เจ้า	206	0.1542	50.3210
111	แห่ง	202	0.1512	50.4722
112	ถูก	199	0.1490	50.6212
113	เครื่อง	197	0.1475	50.7687
114	ส่ง	196	0.1467	50.9154
115	เดียว	196	0.1467	51.0621
116	คิด	194	0.1452	51.2073
117	คำ	194	0.1452	51.3525
118	ลูก	194	0.1452	51.4977
119	แบบ	194	0.1452	51.6429
120	ราช	191	0.1430	51.7859

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
121	เคย	189	0.1415	51.9274
122	ค้า	187	0.1400	52.0674
123	พอ	187	0.1400	52.2074
124	ขอ	186	0.1393	52.3467
125	ลัง	181	0.1355	52.4822
126	สุด	181	0.1355	52.6177
127	เช่น	181	0.1355	52.7532
128	แม่	178	0.1333	52.8865
129	กรรม	177	0.1325	53.0190
130	จริง	177	0.1325	53.1515
131	เรียน	177	0.1325	53.2840
132	ข้อ	176	0.1318	53.4158
133	สามารถ	176	0.1318	53.5476
134	ชาติ	175	0.1310	53.6786
135	บาท	174	0.1303	53.8089
136	ดัง	174	0.1303	53.9392
137	อะไร	171	0.1280	54.0672
138	เท่า	169	0.1265	54.1937
139	รัก	168	0.1258	54.3195
140	รัฐ	168	0.1258	54.4453
141	ติด	167	0.1250	54.5703
142	ประชา	166	0.1243	54.6946
143	หมาย	165	0.1235	54.8181
144	ตำ	163	0.1220	54.9401
145	กล่าว	162	0.1213	55.0614
146	คง	162	0.1213	55.1827
147	ชื่อ	160	0.1198	55.3025
148	ต้น	159	0.1190	55.4215
149	สร้าง	159	0.1190	55.5405
150	ราย	158	0.1183	55.6588

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
151	หลัง	158	0.1183	55.7771
152	แก่	158	0.1183	55.8954
153	สือ	157	0.1175	56.0129
154	ต่าง	156	0.1168	56.1297
155	ดิน	156	0.1168	56.2465
156	ภาพ	155	0.1160	56.3625
157	สอง	155	0.1160	56.4785
158	มหา	154	0.1153	56.5938
159	พวก	154	0.1153	56.7091
160	ต่างๆ	152	0.1138	56.8229
161	ตอน	152	0.1138	56.9367
162	สำหรับ	152	0.1138	57.0505
163	เหมือน	151	0.1131	57.1636
164	พ่อ	150	0.1123	57.2759
165	ควร	148	0.1108	57.3867
166	บริษัท	148	0.1108	57.4975
167	อัน	148	0.1108	57.6083
168	เสียง	148	0.1108	57.7191
169	น้อย	147	0.1101	57.8292
170	ระ	147	0.1101	57.9393
171	ฉัน	146	0.1093	58.0486
172	ร่วม	146	0.1093	58.1579
173	หมด	146	0.1093	58.2672
174	อาจ	146	0.1093	58.3765
175	ดำ	145	0.1086	58.4851
176	ไร	145	0.1086	58.5937
177	ปัญหา	144	0.1078	58.7015
178	กิน	143	0.1071	58.8086
179	ที	143	0.1071	58.9157
180	ด้าน	142	0.1063	59.0220

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
181	น่า	142	0.1063	59.1283
182	หัว	142	0.1063	59.2346
183	เพียง	142	0.1063	59.3409
184	ชาว	141	0.1056	59.4465
185	พูด	141	0.1056	59.5521
186	ธรรม	138	0.1033	59.6554
187	นอก	137	0.1026	59.758
188	ใคร	137	0.1026	59.8606
189	ข่าว	136	0.1018	59.9624
190	ใหม่	135	0.1011	60.0635
191	ครับ	133	0.0996	60.1631
192	มือ	133	0.0996	60.2627
193	โรง	133	0.0996	60.3623
194	ขาย	130	0.0973	60.4596
195	คณะ	129	0.0966	60.5562
196	ตา	129	0.0966	60.6528
197	ยิ่ง	129	0.0966	60.7494
198	ช่วย	127	0.0951	60.8445
199	ร้อย	127	0.0951	60.9396
200	เรือ	127	0.0951	61.0347
201	แสดง	127	0.0951	61.1298
202	สิ่ง	126	0.0943	61.2241
203	กลาง	124	0.0928	61.3169
204	โลก	124	0.0928	61.4097
205	สี	123	0.0921	61.5018
206	สูง	123	0.0921	61.5939
207	เปิด	122	0.0913	61.6852
208	บอก	121	0.0906	61.7758
209	ใบ	121	0.0906	61.8664
210	เอก	120	0.0898	61.9562

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
211	ใด	118	0.0883	62.0445
212	อาหาร	117	0.0876	62.1321
213	เขียน	116	0.0868	62.2189
214	ไฟ	116	0.0868	62.3057
215	รวม	115	0.0861	62.3918
216	ข้าว	114	0.0853	62.4771
217	ผ่าน	114	0.0853	62.5624
218	พิมพ์	114	0.0853	62.6477
219	ไหน	114	0.0853	62.7330
220	เสนอ	113	0.0846	62.8176
221	กอง	112	0.0839	62.9015
222	ขณะ	112	0.0839	62.9854
223	พร้อม	112	0.0839	63.0693
224	เรียก	112	0.0839	63.1532
225	ซื้อ	111	0.0831	63.2363
226	ตรง	111	0.0831	63.3194
227	ห้อง	111	0.0831	63.4025
228	ชอบ	110	0.0824	63.4849
229	ทหาร	110	0.0824	63.5673
230	ถนน	110	0.0824	63.6497
231	หาก	110	0.0824	63.7321
232	ดับ	108	0.0809	63.8130
233	เกี่ยว	108	0.0809	63.8939
234	กำ	107	0.0801	63.9740
235	อยาก	107	0.0801	64.0541
236	แน่	107	0.0801	64.1342
237	แม้	107	0.0801	64.2143
238	แรก	107	0.0801	64.2944
239	จัง	106	0.0794	64.3738
240	ไม้	106	0.0794	64.4532

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
241	แรง	106	0.0794	64.5326
242	บ้าง	105	0.0786	64.6112
243	เนิน	105	0.0786	64.6898
244	นวน	104	0.0779	64.7677
245	สาร	104	0.0779	64.8456
246	อ่าน	104	0.0779	64.9235
247	คำ	103	0.0771	65.0006
248	ถาม	103	0.0771	65.0777
249	ตก	103	0.0771	65.1548
250	ล้าน	103	0.0771	65.2319
251	ราคา	102	0.0764	65.3083
252	หลัก	102	0.0764	65.3847
253	ภาย	101	0.0756	65.4603
254	รูป	101	0.0756	65.5359
255	ใช้	101	0.0756	65.6115
256	กลับ	100	0.0749	65.6864
257	ช่วง	99	0.0741	65.7605
258	นี้	99	0.0741	65.8346
259	ทุน	99	0.0741	65.9087
260	มาณ	99	0.0741	65.9828
261	หนัก	99	0.0741	66.0567
262	ยอม	98	0.0734	66.1303
263	สิน	98	0.0734	66.2037
264	เดือน	98	0.0734	66.2771
265	ทอง	97	0.0726	66.3497
266	นาน	97	0.0726	66.4223
267	นั่ง	97	0.0726	66.4949
268	สอบ	97	0.0726	66.5675
269	ภาค	96	0.0719	66.6394
270	วิทยาลัย	96	0.0719	66.7113

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
271	เฉพาะ	96	0.0719	66.7832
272	ผิด	95	0.0711	66.8543
273	บาล	94	0.0704	66.9247
274	สาย	94	0.0704	66.9951
275	หวัด	93	0.0696	67.0647
276	ชาย	92	0.0689	67.1336
277	บท	92	0.0689	67.2025
278	สม	92	0.0689	67.2714
279	แทน	92	0.0689	67.3403
280	กิจ	91	0.0681	67.4084
281	ผลิต	91	0.0681	67.4765
282	ฝ่าย	91	0.0681	67.5446
283	ถือ	90	0.0674	67.6120
284	เกษตร	90	0.0674	67.6794
285	ชั้น	89	0.0666	67.7460
286	เด็ก	89	0.0666	67.8126
287	เธอ	89	0.0666	67.8792
288	ป่า	88	0.0659	67.9451
289	รวจ	88	0.0659	68.0110
290	เปลี่ยน	88	0.0659	68.0769
291	เล่น	88	0.0659	68.1428
292	แก้	88	0.0659	68.2087
293	ใต้	88	0.0659	68.2746
294	กรม	87	0.0651	68.3397
295	นับ	87	0.0651	68.4048
296	ทรง	87	0.0651	68.4699
297	พื้น	87	0.0651	68.5350
298	หญิง	87	0.0651	68.6001
299	ตลอด	86	0.0644	68.6645
300	ทั่ว	86	0.0644	68.7289

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
301	นั่น	86	0.0644	68.7933
302	เริ่ม	86	0.0644	68.8577
303	เขต	86	0.0644	68.9221
304	กำลัง	85	0.0636	68.9857
305	วิธี	85	0.0636	69.0493
306	เนื่อง	85	0.0636	69.1129
307	เพิ่ม	85	0.0636	69.1765
308	โครง	85	0.0636	69.2401
309	ร้อง	84	0.0629	69.3030
310	ศึกษา	84	0.0629	69.3659
311	เพื่อน	84	0.0629	69.4288
312	เหตุ	84	0.0629	69.4917
313	ชุม	83	0.0621	69.5538
314	ตน	83	0.0621	69.6159
315	สึก	83	0.0621	69.6780
316	ชีวิต	82	0.0614	69.7394
317	ไข	82	0.0614	69.8008
318	กำหนด	81	0.0606	69.8614
319	นอน	81	0.0606	69.9220
320	หนด	81	0.0606	69.9826
321	อย่า	81	0.0606	70.0432
322	ชนิด	80	0.0599	70.1031
323	นะ	80	0.0599	70.1630
324	พรรค	80	0.0599	70.2229
325	ข้าง	79	0.0591	70.2820
326	ขาด	79	0.0591	70.3411
327	รอง	79	0.0591	70.4002
328	เที่ยว	79	0.0591	70.4593
329	เกิน	79	0.0591	70.5184
330	บน	78	0.0584	70.5768

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
331	ระหว่าง	78	0.0584	70.6352
332	มิ	78	0.0584	70.6936
333	ฟัง	78	0.0584	70.7520
334	สำคัญ	78	0.0584	70.8104
335	สมัย	78	0.0584	70.8688
336	เหลือ	78	0.0584	70.9272
337	แหละ	78	0.0584	70.9856
338	จิต	77	0.0576	71.0432
339	ระยะ	77	0.0576	71.1008
340	มอง	77	0.0576	71.1584
341	สถาน	77	0.0576	71.2160
342	แต่ง	77	0.0576	71.2736
343	จัก	76	0.0569	71.3305
344	พล	76	0.0569	71.3874
345	ทราบ	75	0.0562	71.4436
346	สัก	75	0.0562	71.4998
347	ปฏิบัติ	74	0.0554	71.5552
348	ลำ	74	0.0554	71.6106
349	อายุ	74	0.0554	71.6660
350	การณ์	73	0.0547	71.7207
351	พยายาม	73	0.0547	71.7754
352	พบ	73	0.0547	71.8301
353	องค์	73	0.0547	71.8848
354	วัด	73	0.0547	71.9395
355	สุข	73	0.0547	71.9942
356	พัฒนา	72	0.0539	72.0481
357	ขัน	71	0.0532	72.1013
358	กลุ่ม	71	0.0532	72.1545
359	สิ้น	71	0.0532	72.2077
360	ดอก	70	0.0524	72.2601

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
361	พัก	70	0.0524	72.3125
362	นาง	69	0.0517	72.3642
363	พิจารณา	69	0.0517	72.4159
364	สาม	69	0.0517	72.4676
365	กร	68	0.0509	72.5185
366	ทัน	68	0.0509	72.5694
367	ระบบ	68	0.0509	72.6203
368	สภาพ	68	0.0509	72.6712
369	สังคม	68	0.0509	72.7221
370	ศาล	68	0.0509	72.7730
371	อื่น	68	0.0509	72.8239
372	ฉบับ	67	0.0502	72.8741
373	กอบ	67	0.0502	72.9243
374	ตอบ	67	0.0502	72.9745
375	สวน	67	0.0502	73.0247
376	เหล่า	67	0.0502	73.0749
377	ทะ	66	0.0494	73.1243
378	ฟ้า	66	0.0494	73.1737
379	ยืน	66	0.0494	73.2231
380	สหรัฐ	66	0.0494	73.2725
381	เต็ม	66	0.0494	73.3219
382	ประโยชน์	65	0.0487	73.3706
383	ยา	65	0.0487	73.4193
384	ร้าน	65	0.0487	73.4680
385	วิ่ง	65	0.0487	73.5167
386	ศาสตร์	65	0.0487	73.5654
387	โอกาส	65	0.0487	73.6141
388	ใส่	65	0.0487	73.6628
389	วง	64	0.0479	73.7107
390	ร่าง	64	0.0479	73.7586

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
391	พืช	64	0.0479	73.8065
392	ลด	64	0.0479	73.8544
393	ภาษา	64	0.0479	73.9023
394	ยก	64	0.0479	73.9502
395	นา	63	0.0472	73.9974
396	อบ	63	0.0472	74.0446
397	เผย	63	0.0472	74.0918
398	เกาะ	63	0.0472	74.1390
399	ขนาด	62	0.0464	74.1854
400	ทรวง	62	0.0464	74.2318
401	บัง	62	0.0464	74.2782
402	พัน	62	0.0464	74.3246
403	สั่ง	62	0.0464	74.3710
404	เล็ก	62	0.0464	74.4174
405	เทพ	62	0.0464	74.4638
406	ชม	61	0.0457	74.5095
407	หลวง	61	0.0457	74.5552
408	เลือก	61	0.0457	74.6009
409	แก	61	0.0457	74.6466
410	ธนาคาร	60	0.0449	74.6915
411	มัก	60	0.0449	74.7364
412	สน	60	0.0449	74.7813
413	สมัคร	60	0.0449	74.8262
414	ก่อ	59	0.0442	74.8704
415	เสริม	59	0.0442	74.9146
416	ลักษณะ	58	0.0442	74.9580
417	สู่	58	0.0442	75.0014
418	อาจารย์	58	0.0442	75.0448
419	ปาก	57	0.0427	75.0875
420	ปรากฏ	56	0.0419	75.1294

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
421	ตัด	56	0.0419	75.1713
422	พิเศษ	56	0.0419	75.2132
423	สภา	56	0.0419	75.2551
424	สื่อ	56	0.0419	75.2970
425	อื่นๆ	56	0.0419	75.3389
426	เก็บ	56	0.0419	75.3808
427	ปัจจุบัน	55	0.0412	75.4220
428	บุคคล	55	0.0412	75.4632
429	พี่	55	0.0412	75.5044
430	หน่วย	55	0.0412	75.5456
431	หวัง	55	0.0412	75.5868
432	อเมริกา	55	0.0412	75.6280
433	โรค	55	0.0412	75.6692
434	คืน	54	0.0404	75.7096
435	คับ	54	0.0404	75.7500
436	ร้าย	54	0.0404	75.7904
437	รักษา	54	0.0404	75.8308
438	วิชา	54	0.0404	75.8712
439	เล่า	54	0.0404	75.9116
440	เชื่อ	54	0.0404	75.9520
441	เหนือ	54	0.0404	75.9924
442	แนว	54	0.0404	76.0328
443	จ่าย	53	0.0397	76.0725
444	ชั่ว	53	0.0397	76.1122
445	คดี	53	0.0397	76.1519
446	บัญชา	53	0.0397	76.1916
447	ณ	53	0.0397	76.2313
448	ปรับ	53	0.0397	76.2710
449	ร้อน	53	0.0397	76.3107
450	สำนัก	53	0.0397	76.3504

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
451	ไม	53	0.0397	76.3901
452	แข่ง	53	0.0397	76.4298
453	จับ	53	0.0397	76.4687
454	ตี	52	0.0389	76.5076
455	ยื่น	52	0.0389	76.5465
456	สวย	52	0.0389	76.5854
457	สมาคม	52	0.0389	76.6243
458	สำเร็จ	52	0.0389	76.6632
459	กฎ	51	0.0382	76.7014
460	ฐาน	51	0.0382	76.7396
461	ตาย	51	0.0382	76.7778
462	หาย	51	0.0382	76.8160
463	ใกล้	51	0.0382	76.8542
464	คู่	50	0.0374	76.8916
465	ปลูก	50	0.0374	76.9290
466	บริหาร	50	0.0374	76.9664
467	ยาก	50	0.0374	77.0038
468	สาว	50	0.0374	77.0412
469	เภท	50	0.0374	77.0786
470	แค่	50	0.0374	77.1160
471	ลอง	49	0.0367	77.1527
472	วาง	49	0.0367	77.1894
473	ศูนย์	49	0.0367	77.2261
474	เย็น	49	0.0367	77.2628
475	อัตรา	49	0.0367	77.2995
476	เดิม	49	0.0367	77.3362
477	เนื้อ	49	0.0367	77.3729
478	แปลง	49	0.0367	77.4096
479	ไง	49	0.0367	77.4463
480	แผ่น	49	0.0367	77.4830

ตารางที่ ก1 (ต่อ)

ลำดับที่	คำไทย	ความถี่	เปอร์เซ็นต์	เปอร์เซ็นต์สะสม
481	ถ่าย	48	0.0359	77.5189
482	ท่า	48	0.0359	77.5548
483	ทั่ง	48	0.0359	77.5907
484	ทาน	47	0.0352	77.6259
485	มาย	47	0.0352	77.6611
486	หน่อย	47	0.0352	77.6963
487	ศิลป	47	0.0352	77.7315
488	สมาชิก	47	0.0352	77.7667
489	อำเภอ	47	0.0352	77.8019
490	เบี้ย	47	0.0352	77.8371
491	โฆษณา	47	0.0352	77.8723
492	แจ้ง	47	0.0352	77.9075
493	ข้าม	46	0.0344	77.9419
494	คม	46	0.0344	77.9763
495	ครัว	46	0.0344	78.0107
496	มั่น	46	0.0344	78.0451
497	พุทธ	46	0.0344	78.0795
498	รัฐมนตรี	46	0.0344	78.1139
499	สัมพันธ์	46	0.0344	78.1483
500	วิทยุ	46	0.0344	78.1827
501	เสร็จ	46	0.0344	78.2171
502	อาทิตย์	46	0.0344	78.2515
503	เร็ว	46	0.0344	78.2859
504	ตะ	45	0.0337	78.3196
505	ครู	45	0.0337	78.3533
506	ข้า	45	0.0337	78.3870
507	กาย	45	0.0337	78.4207
508	แผน	45	0.0337	78.4544
509	ปืน	44	0.0329	78.4873
510	กลาย	44	0.0329	78.5202
511	ธุรกิจ	44	0.0329	78.5531

ตารางที่ ก2 รหัสแอสกีของสำนักงานมาตรฐานผลิตภัณฑ์อุตสาหกรรม (ส.ม.อ.)

Hex Digits	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0				0	@	P	`	p				ฐ	ภ	◌ะ	เ-	๐
-1			!	1	A	Q	a	q			ก	ฑ	ม	◌ั	แ-	๑
-2			"	2	B	R	b	r			ข	ฒ	ย	◌า	โ-	๒
-3			#	3	C	S	c	s			ฃ	ณ	ร	◌ำ	ใ-	๓
-4			$	4	D	T	d	t			ค	ด	ฤ	◌ิ	ไ-	๔
-5			%	5	E	U	e	u			ฅ	ต	ล	◌ี	◌ๅ	๕
-6			&	6	F	V	f	v			ฆ	ถ	ฦ	◌ึ	ๆ	๖
-7			'	7	G	W	g	w			ง	ท	ว	◌ื	◌็	๗
-8			(	8	H	X	h	x			จ	ธ	ศ	◌ุ	◌่	๘
-9			)	9	I	Y	i	y			ฉ	น	ษ	◌ู	◌้	๙
-A			*	:	J	Z	j	z			ช	บ	ส		◌๊	๚
-B			+	;	K	[	k	{			ซ	ป	ห		◌๋	๛
-C			,	<	L	\	l	|		ฌ	ผ	ฬ		◌์	
-D			-	=	M	]	m	}			ญ	ฝ	อ		◌ ◌	
-E			.	>	N	^	n	~			ฎ	พ	ฮ		◌ ◌	
-F			/	?	O	_	o				ฏ	ฟ	ฯ	฿	◌	

ภาคผนวก ข

ชีวิตเรา คือของรักของหวงที่สุดในชีวิตเขา
วิธีตอบแทนบุญคุณพ่อแม่ที่ดีที่สุด คือ
การดูแลตัวเองให้ดีที่สุด
หลายคนชอบพูดว่า
โตขึ้นจะตอบแทนบุญคุณพ่อแม่ ด้วยการเลี้ยงดูพ่อแม่
หาเงินให้พ่อแม่ใช้เยอะๆ ไม่ให้ท่านลำบาก
ฉันกลับมองว่า
เราจะตอบแทนบุญคุณพ่อแม่..ไม่เห็นต้องรอให้โต
ทุกขั้นตอนชีวิตของเรา
เราสามารถทำได้ตลอดเวลา
นั่นคือ
การดูแลชีวิตตัวเองให้ดีที่สุด
ให้สมกับที่เขาดูแลเรามาตั้งแต่เล็กๆ
การที่จะทำให้เขาภูมิใจ
คือการทำให้เขาวางใจ เบาใจ ไว้ใจ
ไม่ทำตัวเป็นภาระ
ทำให้เขาเห็นว่า เราดูแลตัวเองได้ดี
เมื่อไหร่ที่พ่อแม่เลิกบ่น
เรื่องห่วงคงเลิกไม่ได้ ยังไงก็ต้องห่วง
แต่หากเขาวางใจ ไม่ว่าเราจะทำอะไรที่ไหน
เขาก็มั่นใจในตัวเรา
ไม่ต้องอยู่ในความดูแลของเขาอย่างใกล้ชิดอยู่ตลอดเวลา
ไปไหนมาไหนได้โดยเขาไม่วิตกกังวล
และตัดสินใจได้ด้วยตัวเอง
ฉันถือว่า นั่นแหละคือการตอบแทนบุญคุณที่ดีที่สุดแล้ว
ชีวิตเราคือของรักของหวงที่สุดในชีวิตเขา
หากเราดูแลได้เป็นอย่างดี
มีหรือ ที่วันหนึ่ง
เขาจะไม่ยกให้เราดูแลอย่างสมบูรณ์

รูปที่ ข1 ตัวอย่างข้อมูล a4.txt ที่ใช้ในการทดสอบ

ตั้งสถาบันทดสอบการศึกษาเป็นรูปร่างคาดทดสอบพื้นฐานปี48

ดร.สิริกร มณีรินทร์ รมช.ศึกษาธิการ ในฐานะที่ปรึกษาคณะกรรมการเตรียมการจัดตั้งสถาบันทดสอบทางการศึกษาแห่งชาติ (สทก.) เปิดเผยภายหลังการประชุมคณะกรรมการเตรียมการจัดตั้งฯ เมื่อวันที่ 12 ก.พ. ว่า ที่ประชุมได้รับทราบรายงานจาก ดร.สุชาติ เมืองแก้ว รองเลขาธิการคณะกรรมการการอุดมศึกษา (กอ.) ในฐานะประธานคณะกรรมการพัฒนาระบบกลางการรับนิสิตนักศึกษาเข้าศึกษาในสถาบันอุดมศึกษา หรือแอดมิชชั่น ในปีการศึกษา 2549 ที่เป็นผู้ศึกษาเกี่ยวกับรูปแบบการดำเนินงานระบบแอดมิชชั่น และได้ข้อสรุปว่าให้จัดตั้งองค์กรกลางระดับประเทศที่เป็นหน่วยงานอิสระไม่เป็นส่วนราชการ โดยให้มีคณะกรรมการบริหารที่มาจากสถาบันอุดมศึกษาต่าง ๆ เพื่อทำหน้าที่ประสานงานการสมัคร และแจ้งผลระหว่างนักเรียนกับสถาบันอุดมศึกษา นอกจากนี้เพื่อให้รูปแบบการทำงานสะดวก และคล่องตัว ควรให้จัดตั้งหน่วยงานระดับภูมิภาคทั้ง 5 ภาค ได้แก่ ภาคเหนือ ภาคกลาง ภาคใต้ ภาคตะวันออก และภาคตะวันออกเฉียงเหนือ เพื่อเป็นศูนย์เครือข่ายในการรับส่งเอกสาร การส่งข้อมูลทางอิเล็กทรอกนิกส์ และประสานการแจ้งผลไปยังมหาวิทยาลัยในภูมิภาค นั้น ๆ ด้วย ซึ่งตนจะนำข้อสรุปดังกล่าวเสนอต่อที่ประชุมอธิการบดีแห่งประเทศไทย (ทปอ.) ในวันที่ 28 ก.พ. นี้ เพื่อพิจารณาว่าจะให้หน่วยงานใดทำหน้าที่ดังกล่าว ซึ่งอาจจะเป็นสำนักงานคณะกรรมการการอุดมศึกษา (สกอ.) หรือ ทปอ. ก็ได้ อย่างไรก็ตามจากการไปดูงานการคัดเลือกบุคคลเข้าศึกษาต่อในสถาบันอุดมศึกษาของประเทศอังกฤษ พบว่าในระยะยาวนั้นสถาบันทดสอบทางการศึกษาแห่งชาติจะต้องคำนึงถึงความเชื่อมโยงของการประเมินผลให้กับอาชีวศึกษาและคนที่อยู่นอกระบบโรงเรียนด้วย

รศ.ดร.คุณหญิง สุมนฑา พรหมบุญ ประธานคณะกรรมการเตรียมการจัดตั้งฯ กล่าวว่า คณะกรรมการจัดทำมาตรฐานการศึกษาขั้นพื้นฐานด้านผู้เรียน ซึ่งมี ศ.ดร.สมหวัง พิธิยานุวัฒน์ ผอ.สำนักงานรับรองมาตรฐานและประเมินคุณภาพการศึกษา (สมศ.) เป็นประธาน ได้ยืนยันกับที่ประชุมว่าจะพิจารณาเกณฑ์มาตรฐานดังกล่าวให้เสร็จในเดือนพฤษภาคม 47 นี้ จากนั้นจะจัดทำข้อสอบให้เสร็จก่อนที่จะมีการทดสอบในเดือนกุมภาพันธ์ 2548 โดยการทดสอบจะต้องเป็นไปตามเกณฑ์มาตรฐานเดียวกันทั่วประเทศ นอกจากนี้ที่ประชุมยังได้อนุมัติจัดตั้งคณะกรรมการอำนวยการทดสอบระดับชาติ คณะกรรมการดำเนินการทดสอบระดับชาติ และคณะกรรมการดำเนินการระดับเขตพื้นที่การศึกษาเพื่อมาทำงานในสถาบันทดสอบการศึกษาแห่งชาติอีกด้วย.

รูปที่ ข2 ตัวอย่างข้อมูล b1.txt ที่ใช้ในการทดสอบ

ประวัติผู้เขียน

นายคัมภีร์ เสริมกวินรักษ์ เกิดเมื่อวันที่ 29 มกราคม พ.ศ. 2524 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาวิทยาศาสตรบัณฑิต (คณิตศาสตร์ประยุกต์) จากคณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2544 ได้เข้าศึกษาระดับปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิทยาการคอมพิวเตอร์ ในปี พ.ศ. 2545