

การแบ่งคำภาษาไทยด้วยเทคนิคไตรแกรม  
**Trigram Technique in Thai Word Segmentation**  
ประเภทโครงการ การแบ่งคำไทย

รายงานฉบับสมบูรณ์

เสนอต่อ

ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ  
สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ  
กระทรวงวิทยาศาสตร์และเทคโนโลยี

ได้รับทุนอุดหนุนโครงการวิจัย พัฒนาและวิศวกรรม  
โครงการการแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย ครั้งที่ 11  
ประจำปีงบประมาณ 2551

โดย

น.ส. เกศราภรณ์ ชื้อสัตย์พานิชย์

อาจารย์ที่ปรึกษาโครงการ

อ.ดร.อดิวงค์ สุชาติ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

## กิตติกรรมประกาศ

ขอกราบขอบพระคุณบิดา มารดา และบุคคลในครอบครัว รวมถึงเพื่อนๆ ทุกคนที่ให้กำลังใจและให้ความสนับสนุนให้การจัดทำโครงการเป็นไปอย่างราบรื่น โดยเฉพาะอย่างยิ่งขอขอบคุณท่านอาจารย์ อติวงศ์ สุชาโต อาจารย์ที่ปรึกษาที่ให้คำปรึกษา คำแนะนำที่ดีและกำลังใจทำให้โครงการนี้สำเร็จได้ตามความตั้งใจ รวมถึงขอขอบคุณห้องปฏิบัติการวิจัยระบบภาษาพูดของคณะวิศวกรรมศาสตร์ที่เอื้อเพื่อสถานที่ในการทำงาน และเพื่อนๆ สมาชิกในห้องปฏิบัติการที่รับฟังแนวคิดและให้คำแนะนำต่างๆ ตลอดมา

สุดท้ายนี้ขอขอบคุณศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ และสำนักงานส่งเสริมอุตสาหกรรมซอฟต์แวร์แห่งชาติที่ให้โอกาสและมอบทุนอุดหนุนโครงการการแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย ครั้งที่ 11 ในการพัฒนาโครงการการแบ่งคำภาษาไทยด้วยเทคนิคไตรแกรมนี้

เกศราภรณ์ ชู่อัสตัยพาณิชย์

## บทคัดย่อ

โครงการการแบ่งคำภาษาไทยด้วยเทคนิคไตรแกรมเป็นการพัฒนาโปรแกรมแบ่งคำไทยเพื่อแก้ปัญหาขอบเขตที่ไม่ชัดเจนของคำและความกำกวมของภาษาไทย ที่สามารถนำไปใช้ประโยชน์ในด้านอื่นๆ ที่เกี่ยวข้องในการประมวลผลภาษาไทยได้ เช่น การปรับรูปเขียนให้เป็นรูปอ่านในระบบแปลงรูปอักษรเป็นเสียงพูด (Text-to-Speech) ระบบการแปลภาษาไทยเป็นภาษาต่างประเทศ การสกัดคำสำคัญ เพื่อใช้ในระบบการค้นคืนข้อมูลสารสนเทศ (Information Retrieval) โดยโปรแกรมจะถูกพัฒนาด้วยเทคนิคไตรแกรมซึ่งเป็นวิธีการนำสถิติมาช่วยในการเลือกแบ่งขอบเขตคำ

## **Abstract**

Trigram Technique in Thai Word Segmentation Project is developed for solving unclear boundary and ambiguous Thai word. Thai Word Segmentation is used in Thai Language Processing such as Text-to-Speech, Machine Translation, Word Extraction in Information Retrieval. Thai Word Segmentation is developed with Trigram technique base on statistic framework.

## บทนำ

### แนวคิด ความสำคัญ และความเป็นมาของโครงการ

วิทยาการคอมพิวเตอร์ได้มีกระบวนการประมวลผลภาษาธรรมชาติของมนุษย์ (Natural Language Processing) ไม่ว่าจะเป็นการปรับรูปเขียนให้เป็นรูปอ่านในระบบแปลงรูปอักษรเป็นเสียงพูด (Text-to-Speech) ระบบการแปลภาษาไทยเป็นภาษาต่างประเทศ การสกัดคำสำคัญเพื่อใช้ในการระบบการค้นคืนข้อมูลสารสนเทศ (Information Retrieval) การตรวจไวยากรณ์ในโปรแกรมเอกสาร (Check Grammar) ซึ่งในการประมวลผลภาษาธรรมชาตินั้นจะมีกระบวนการที่สำคัญขั้นตอนหนึ่งที่เป็นขั้นตอนแรกคือ การแบ่งคำ แต่เนื่องจากภาษาไทยเป็นภาษาที่ไม่มีขอบเขตของคำที่ชัดเจนและยังมีความกำกวมอยู่ในหลายๆ คำ ไม่เหมือนภาษาต่างประเทศอื่น เช่น ภาษาอังกฤษ ที่มีเว้นวรรคเป็นตัวแบ่งได้ว่าส่วนใดคือคำ ทำให้ประเทศไทยได้มีการทำวิจัยด้านนี้มาอย่างต่อเนื่องเป็นเวลาหลายปี และจนถึงปัจจุบันยังมีการวิจัยแบ่งคำภาษาไทย โดยต่อยอดงานวิจัยเดิมหรือค้นหาวิธีการใหม่ๆ เพื่อให้การแบ่งคำภาษาไทยมีความถูกต้องมากยิ่งขึ้น

การวิจัยในเรื่องแบ่งคำที่ผ่านมายังพบปัญหาที่ทำให้ไม่สามารถนำวิธีการที่ถูกนำเสนอไปใช้งานได้ ในภาคธุรกิจ ทำให้การวิจัยแบ่งคำยังไม่ถือว่าก้าวหน้าเท่าที่ควร โดยมีสาเหตุหลักคือ

1. ไม่มีมาตรฐานของการแบ่งคำไทย
2. ไม่มีข้อมูลด้านการแบ่งคำไทย
3. ไม่มีมาตรฐานเดียวกันในการวัดเปรียบเทียบความสามารถของการแบ่งคำไทย

จากสาเหตุดังกล่าวทำให้ผู้พัฒนาได้เข้าใจถึงปัญหาที่ควรทำการค้นหาวิธีและพัฒนา โปรแกรมแบ่งคำภาษาไทยให้มีมาตรฐานการแบ่งคำไทยแบบเดียวกันและให้มีการเผยแพร่งานวิจัยต่อวงการวิจัยภาควิทยาการคอมพิวเตอร์ต่อไป เพื่อให้เป็นประโยชน์ต่อการใช้งานจริงและสามารถนำไปติดต่อขอได้ในอนาคต

## สารบัญ

หน้า

กิตติกรรมประกาศ.....	2
บทคัดย่อ.....	3
Abstract.....	4
บทนำ.....	5
วัตถุประสงค์และเป้าหมาย.....	7
รายละเอียดของการพัฒนา.....	8
กลุ่มผู้ใช้โปรแกรม.....	12
ผลการทดสอบ.....	13
สรุปและข้อเสนอแนะ.....	13
เอกสารอ้างอิง.....	14
ภาคผนวก.....	15
คู่มือการติดตั้ง.....	15
คู่มือการใช้งาน.....	18

## วัตถุประสงค์และเป้าหมาย

### วัตถุประสงค์

เพื่อพัฒนาโปรแกรมแบ่งคำภาษาไทย

### เป้าหมาย

1. ได้พัฒนาโปรแกรมแบ่งคำไทยให้มีความถูกต้องมากยิ่งขึ้น
2. เพื่อให้การคิดค้นและการพัฒนาโปรแกรมแบ่งคำนี้เกิดประโยชน์ต่อการนำไปใช้งานได้จริง หรือนำไปพัฒนาต่อยอดได้ในอนาคต

## รายละเอียดของการพัฒนา

คำภาษาไทยส่วนมากจะมีความกำกวม เนื่องจากการผสมคำนั้นไม่มีหลักเกณฑ์ที่ชัดเจนแน่นอน ทำให้มีแบบในการแบ่งคำได้หลายแบบ จึงทำการพัฒนาโปรแกรมด้วยเทคนิคไตรแกรมโมเดลมาสำหรับหาค่าความน่าจะเป็นเพื่อใช้เป็นคะแนนของแต่ละแบบ ซึ่งเทคนิคไตรแกรมโมเดลจำเป็นต้องใช้คลังข้อความที่เก็บข้อความที่ถูกแบ่งคำไว้แล้วขนาดใหญ่ โดยจะนำ BEST Corpus ที่จัดทำขึ้นโดยเนคเทคไว้ประมาณ 5 ล้านคำมาใช้

การแบ่งคำนี้จะนำสถิติเข้ามาใช้แก้ปัญหาการแบ่งคำ โดยนำโมเดลไตรแกรมเข้ามาคำนวณค่าความน่าจะเป็นของประโยคซึ่งคำนวณได้ตามสมการ ดังนี้

$$\begin{aligned} P(W) &= \prod_{i=1}^n P(w_{i,n}) \\ &= \prod_{i=1}^n P(w_i | w_{i-1}, w_{i-2}) \end{aligned}$$

จากสมการเป็นการคำนวณหาค่าความน่าจะเป็นของแต่ละประโยค โดย  $W$  คือประโยคที่ถูกแบ่งมาแล้ว และประโยค  $W$  จะประกอบด้วยคำต่างๆ ซึ่ง  $W = w_1 w_2 \dots w_n$  โดย  $w_i$  คือคำศัพท์และการคำนวณค่าความน่าจะเป็นของแต่ละประโยคจะมีสมมติฐานว่าความน่าจะเป็นของ  $w_i$  จะขึ้นอยู่กับ  $w_{i-1}$  และ  $w_{i-2}$  เท่านั้น

แต่เนื่องจากการคำนวณตามสมการนี้ จะต้องใช้คลังข้อความขนาดใหญ่ที่ควรจะมีมากกว่า  $n^3$  โดยที่  $n$  คือจำนวนคำที่เป็นไปได้ทั้งหมด สาเหตุที่วิธีนี้ต้องใช้คลังข้อความที่มีขนาดมากกว่า  $n^3$  คำ เนื่องจากวิธีนี้ต้องมีการนำคำสถิติการเกิดของคำ 3 คำที่ติดกันมาใช้ในการคำนวณ ดังนั้นเพื่อให้มีคำสถิติของการเกิดคำ 3 คำที่ติดกันทุกๆ แบบ อย่างน้อยที่สุดจะต้องใช้  $n^3$  คำ จึงนำความน่าจะเป็นของไบนแกรมและยูนิแกรมเข้ามาช่วยในการคำนวณด้วย (อ้างอิงจากงานวิจัยเรื่อง "A Statistical Approach to Thai Word Filtering" ของอัศนีย์ ก่อตระกูล) โดยใช้สมการ ดังนี้

$$\begin{aligned} \prod_{i=1}^n P(w_i | w_{i-2}, w_{i-1}) &= \prod_{i=1}^n (\lambda_1 P(w_n) + \lambda_2 P(w_n | w_{n-1}) \\ &\quad + \lambda_3 P(w_n | w_{n-1}, w_{n-2})) \end{aligned}$$

ค่า  $\lambda_1, \lambda_2, \lambda_3$  ให้มีค่าเท่ากับ 0.1, 0.3, 0.6 ตามลำดับ



เนื่องจากอาจจะไม่มีการเกิดขึ้นของคำที่ติดกันตามแบบที่ต้องการหาค่าความน่าจะเป็นในคลังข้อความ จึงทำการเพิ่ม 1 / จำนวนคำทั้งหมด บวกเข้าไปในสมการข้างต้นด้วย

การแบ่งคำมีขั้นตอนดังนี้

### 1. แบ่งคำทุกแบบที่เป็นไปได้

นำ Input ที่เป็น Text File มาอ่านทีละบรรทัด แล้วนำแต่ละบรรทัดนั้นมาแบ่งตามการเว้นช่องว่างของข้อความ จากนั้นจะนำแต่ละส่วนข้อความมาทำการหาแบบการแบ่งคำโดย Lookup คำ ซึ่งคำได้ถูกเก็บไว้ในฐานข้อมูล (MySQL) ซึ่งในขั้นตอนหาแบบคำที่เป็นไปได้ จะมีการตัดทิ้งแบบคำที่มีค่ารวมความน่าจะเป็นต่ำกว่า Threshold ที่กำหนดไว้ ซึ่ง Threshold คือ

$$\text{Threshold} = r * \text{THESHO}$$

โดย r คือจำนวนคำที่ถูกแบ่ง และ THESHO กำหนดไว้ 4 ค่า ดังนี้

$$\text{THESHO1} = (0.6 * 0.0004) + (0.3 * 0.0005) + (0.1 * 0.0006) + \text{NOORDER}$$

$$\text{THESHO2} = 0.6 * 0.0003 + (0.3 * 0.0004) + (0.1 * 0.0005) + \text{NOORDER}$$

$$\text{THESHO3} = (0.6 * 0.0002) + (0.3 * 0.0003) + (0.1 * 0.0004) + \text{NOORDER}$$

$$\text{THESHO4} = (0.6 * 0.0001) + (0.3 * 0.0002) + (0.1 * 0.0003) + \text{NOORDER}$$

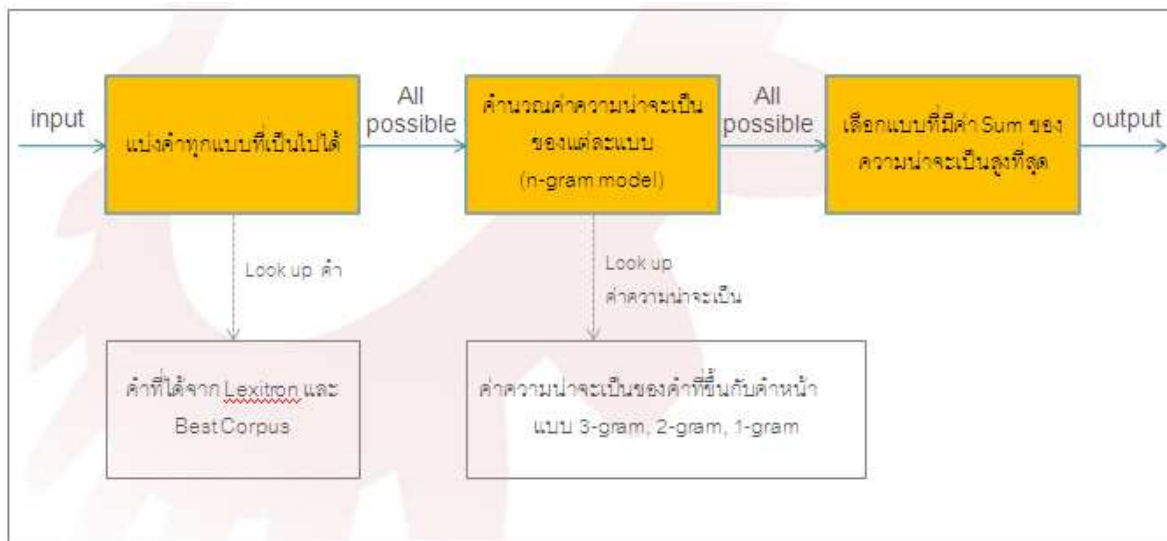
โดย NOORDER คือ คำ 1 / จำนวนคำทั้งหมด

ที่กำหนดค่า Threshold ไว้ 4 ค่าเนื่องจากการตัดทิ้งตาม Threshold อาจจะทำให้ไม่เหลือแบบใดๆ เลยเนื่องจากค่า Threshold อาจสูงเกินไป จึงทำการเช็คถ้าตัดทิ้งตาม Threshold ที่สูงสุดแล้วไม่เหลือแบบใดๆ เลย ก็จะใช้ค่า Threshold ที่สูงรองลงมาเรื่อยๆ จนกว่าจะพบแบบที่เป็นไปได้อย่างน้อยหนึ่งแบบ แต่ถ้าใช้ Threshold ที่ต่ำที่สุดแล้วยังไม่เหลือแบบใดๆ เลย ก็จะไม่ทำการตัดทิ้งตาม Threshold จะทำให้เหลือแบบที่เป็นไปได้ทุกแบบ

### 2. หาค่าความน่าจะเป็นทุกแบบที่ได้จากข้อ 1

นำทุกแบบที่แบ่งได้มาหาค่ารวมผลคูณความน่าจะเป็น โดยการหาจะไม่ได้คำนวณทุกแบบจนจบแล้วจึงเลือก แต่ในระหว่างการหาค่าความน่าจะเป็นและรวมผลคูณความน่าจะเป็นหรือ Search จะทำการเลือกแบบที่มีค่าผลรวมแต่ละ Node ที่สูงสุด จำนวน m แบบ โดยจำนวน m คือ จำนวนอักขระ/10

### 3. เลือกแบบแบ่งคำที่มีผลความน่าจะเป็นสูงที่สุด



รูปที่ 1 ขั้นตอนการแบ่งคำ

โปรแกรมเป็นลักษณะ Command Line ซึ่งจะรับ Input เป็น Text File เข้ารหัสแบบ UTF-8 และเมื่อโปรแกรมแบ่งคำแล้วได้ Output เป็น Text File ที่เข้ารหัสแบบ UTF-8 เช่นเดิม

ในการใช้งาน โปรแกรม เมื่อทำการรัน โปรแกรมแบ่งคำ โปรแกรมจะขึ้นข้อความ Enter in/out file:

ให้พิมพ์คำสั่งด้วยรูปแบบ ดังนี้

-i {input file} -o {output file}

โดย input file คือ Path และชื่อ Text File ที่ต้องการแบ่งคำ และ output file คือ Path และชื่อ Text File ที่เป็นผลลัพธ์ข้อความที่ถูกแบ่งคำแล้ว โดยจะแทรก | หลังคำที่ถูกแบ่งเท่านั้น (ดูรายละเอียดเพิ่มเติมจากคู่มือการใช้งานในภาคผนวก)

```

Palmy V.1.0
Loading training set 1 of 8...
1-gram size = 99274
Loading training set 2 of 8...
Loading training set 3 of 8...
2-gram size = 781881
Loading training set 4 of 8...
Loading training set 5 of 8...
Loading training set 6 of 8...
Loading training set 7 of 8...
Loading training set 8 of 8...
3-gram size = 2388852
Enter in/out file:
  
```

## รูปที่ 2 โปรแกรมการตัดคำภาษาไทยด้วยเทคนิคไตรแกรม

เครื่องมือที่ใช้ในการพัฒนา มีดังนี้

1. ภาษาที่ใช้เขียน: Java
2. API: Java API
3. IDE: NetBeans
4. Training Set Storage: XML File

ขอบเขตและข้อจำกัดของโปรแกรม

1. โปรแกรมพัฒนาด้วยหลักแบ่งคำ“หน่วยเล็กที่สุดที่มีองค์ประกอบความเป็นคำครบถ้วน”
2. โปรแกรมไม่ตรวจสอบความผิดพลาดของการสะกดคำผิดของ Input ที่เข้ามา
3. โปรแกรมแบ่งข้อความจาก Input File ที่เป็น Text File (.txt) ทีละ 1 Text File

## กลุ่มผู้ใช้โปรแกรม

1. ผู้ที่ต้องการแบ่งคำภาษาไทยเพื่องานอื่น เช่น งานในการประมวลผลธรรมชาติ
2. ผู้ที่ต้องการศึกษาเรื่องการแบ่งคำภาษาไทย

## ผลการทดสอบ

ทำการทดสอบ โปรแกรมเพื่อหาค่าความถูกต้องในการแบ่งคำภาษาไทย ด้วยการใช่วิธี F-Measure โดยกำหนดชุดสำหรับทดสอบเป็น ไฟล์ข้อความประเภทบทความจำนวน 85 ไฟล์ เป็นจำนวนค่าประมาณ 5 แสนคำ ซึ่งชุดข้อมูลเหล่านี้ไม่ได้เป็นชุดข้อมูลเดียวกับชุดข้อมูลที่ใช้เป็น Training Set

จากการทดสอบหาค่า F-Measure โดยเฉลี่ยได้ประมาณ 95.50%

## สรุปและข้อเสนอแนะ

จากการพัฒนาและทดสอบโปรแกรมเพื่อให้ได้ความถูกต้องมากขึ้นนั้น ปัจจัยที่มีผลต่อค่าความถูกต้องที่ได้ มีดังนี้

1. การ Prunning เส้นทาง หรือการตัดแบบการแบ่งคำที่มีค่าความน่าจะเป็นต่ำทิ้ง ในระหว่างการหาแบบที่เป็นไปได้

ในระหว่างการหาแบบการแบ่งคำที่เป็นไปได้นั้น โปรแกรมไม่สามารถใช้ทุกแบบการแบ่งที่เป็นได้มาคำนวณทั้งหมด เนื่องจากจะพบแบบที่เป็นไปได้มากมาย ที่มีบางแบบไม่จำเป็นที่จะต้องนำมาคำนวณต่อ โดยมีสมมติฐานว่าแบบที่มีค่าความเป็นไปได้ต่ำมากจะไม่ใช้แบบที่แบ่งคำถูกต้อง จึงจำเป็นต้องทำการตัดแบบเหล่านั้นทิ้งไป โดยมีการกำหนดค่าพารามิเตอร์ หรือค่าขีดแบ่ง เพื่อกำหนดว่าค่าความน่าจะเป็นที่ต่ำกว่าค่าขีดแบ่ง จะต้องถูกตัดทิ้ง ทำให้ค่าพารามิเตอร์เป็นค่าที่สำคัญที่จะต้องเป็นค่าที่เหมาะสมเพื่อไม่ให้เกิดกรณีที่กำหนดค่าขีดแบ่งสูงเกินไป เพราะจะทำให้แบบที่ถูกต้องถูกตัดทิ้งไปด้วย ซึ่งค่าขีดแบ่งที่ยิ่งสูงก็จะยิ่งทำให้โปรแกรมทำงานได้เร็วขึ้น แต่ก็จะสวนทางกับค่าความถูกต้องที่จะทำให้ต่ำลง ดังนั้นความเร็วของเครื่องคอมพิวเตอร์จึงมีผลต่อความถูกต้องของการแบ่งคำด้วย

2. คำที่ไม่ปรากฏในพจนานุกรมและคลังข้อความ

เมื่อพบคำที่ไม่ปรากฏในพจนานุกรมและคลังข้อความ จะทำให้การแบ่งคำผิดพลาดได้ ซึ่งจำเป็นต้องหาวิธีการในการจัดการกับคำเหล่านี้

3. แบบคำที่ติดกันที่ไม่ได้ปรากฏอยู่ในคลังข้อความ Training Set

เมื่อพบแบบคำที่ติดกันที่ไม่ได้ปรากฏอยู่ในคลังข้อความ Training Set ทำให้ค่าความน่าจะเป็นมีน้อยกว่าที่ควรจะเป็น จึงต้องมีวิธีการทำ Smoothing หลากหลายวิธีที่ต่างกันไป โดยจำเป็นต้องเลือกใช้ Smoothing ที่มีเหมาะสมกับคลังข้อความที่ใช้หรือปัจจัยอื่นๆ

## เอกสารอ้างอิง

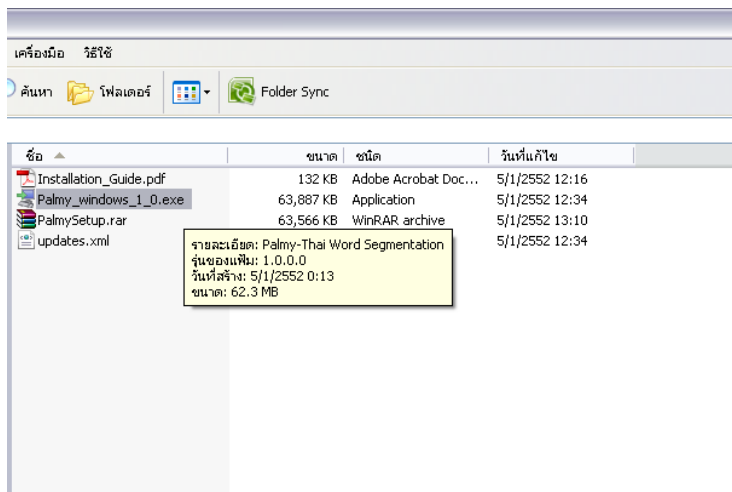
- [1] Kawtrakul, A. et al. 1997, "A statistical approach to Thai word filtering", in The second symposium on natural language processing, Bangkok.
- [2] Stanley F. Chen and Joshua Goodman. 1998, "An Empirical Study of Smoothing Techniques for Language Modeling", Technical Report TR-10-98, Computer Science Group, Harvard University, 1998.

## ภาคผนวก

### คู่มือการติดตั้ง

โปรแกรมแปลคำภาษาไทย มีขั้นตอนการติดตั้งเพื่อใช้งาน ดังนี้

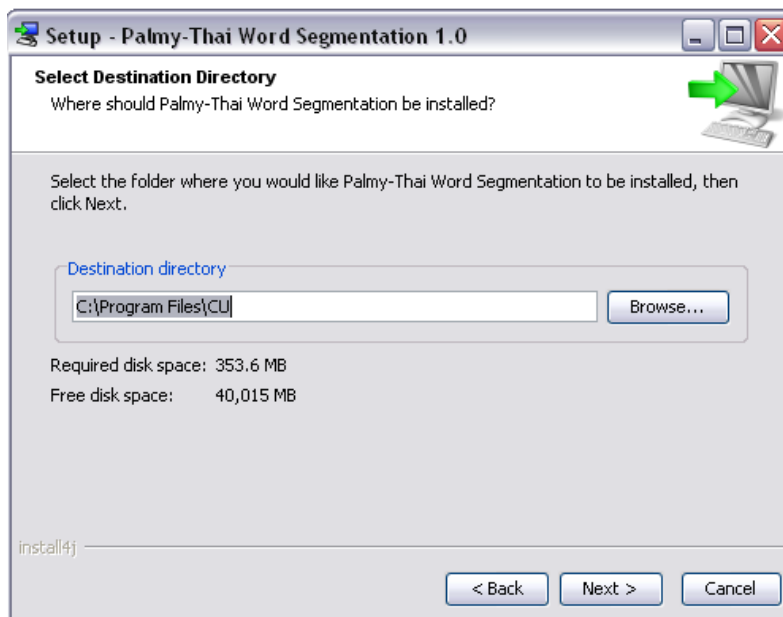
1. ให้ดับเบิลคลิกที่ไฟล์ Palmy\_windows\_1\_0.exe ที่อยู่ในโฟลเดอร์ Palmy1.0 Installer



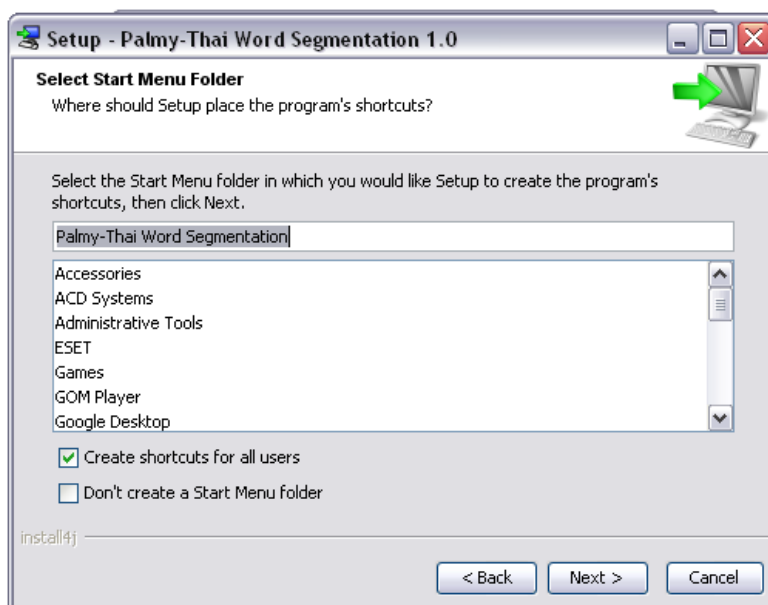
2. จะปรากฏหน้าต่าง Setup – Palmy-Thai Word Segmentation 1.0 จากนั้นคลิกที่ปุ่ม Next



3. เลือก Directory ที่ต้องการลงโปรแกรม โดยเริ่มแรกจะมีค่า Default เป็น C:\Program Files\CU ให้จากนั้นคลิก Next >

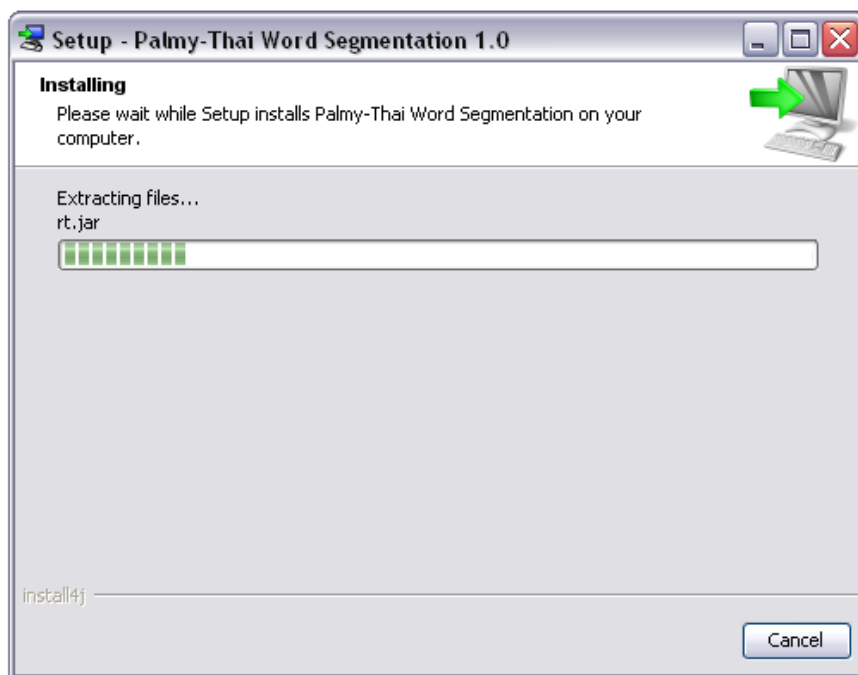


4. เลือกโฟลเดอร์ Start Menu ตามที่ต้องการจากนั้นคลิก Next >

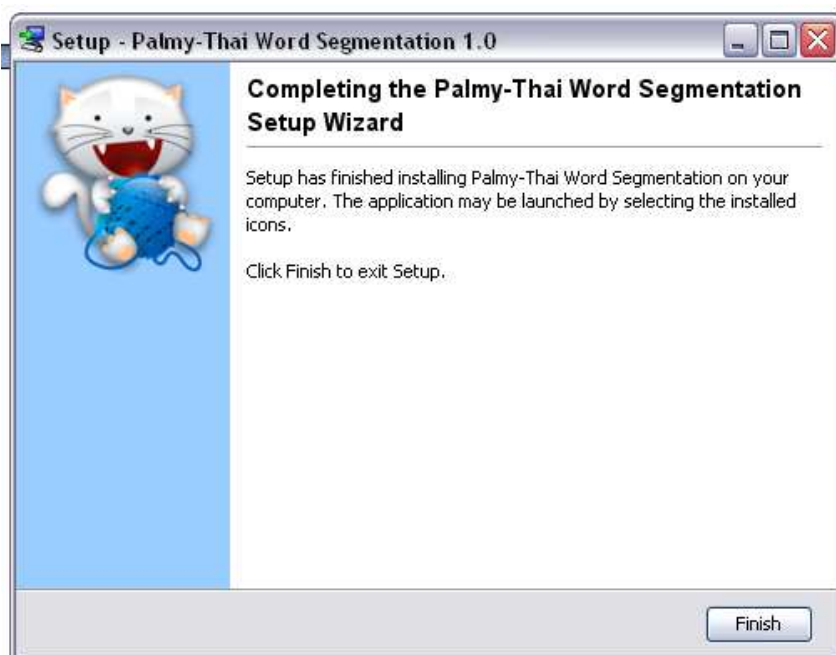


5. จะทำการ Install โปรแกรม



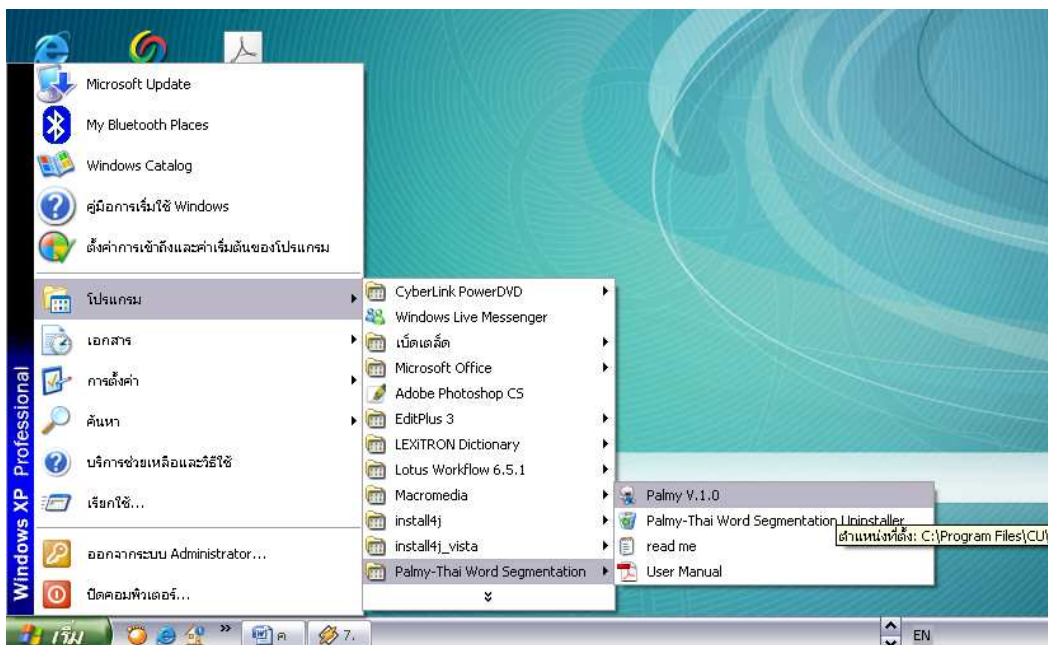


6. เมื่อเสร็จสิ้นสมบูรณ์ จะปรากฏหน้าต่างแสดงข้อความ Completing จากนั้นให้คลิกที่ปุ่ม Finish

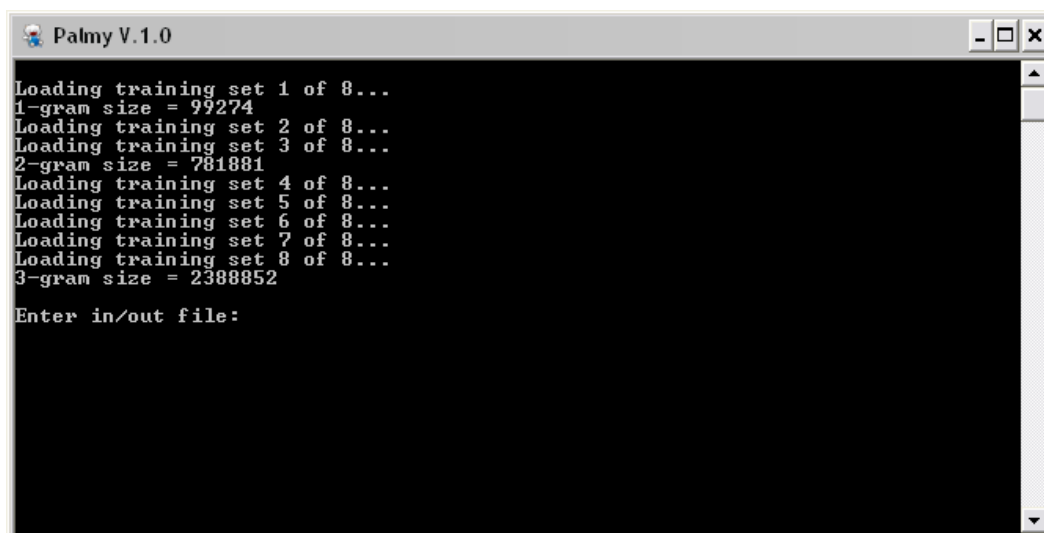


วิธีการใช้งานโปรแกรมแบ่งคำภาษาไทย มีดังนี้

1. เปิดโปรแกรมโดยไปคลิกที่ Start (เริ่ม) > Programs (โปรแกรม) > Palmy-Thai Word Segmentation > Palmy V.1.0



2. จะปรากฏโปรแกรมรูปแบบ Command Line ซึ่งโปรแกรมจะทำการโหลดข้อมูลจากนั้นจะแสดงข้อความ Enter in/out file: พร้อมใช้งาน



3. ให้ทำการพิมพ์คำสั่งเพื่อแบ่งคำด้วยไฟล์รับเข้าและออกที่ต้องการ และกดปุ่ม Enter โดยรูปแบบคำสั่ง ดังนี้

`-i {input file} -o {output file}`

โดย input file คือ Directory และชื่อ Text File ที่ต้องการแบ่งคำ และ output file คือ Directory และชื่อ Text File ที่เป็นผลลัพธ์ข้อความที่ถูกแบ่งคำแล้ว

*ตัวอย่างคำสั่ง*

ต้องการแบ่งคำที่ไฟล์ TEST\_100K.txt ซึ่งอยู่ใน Directory d:/text/ และให้ผลลัพธ์ที่ได้แบ่งคำแล้วอยู่ในไฟล์ o\_TEST\_100K.txt ซึ่งอยู่ใน Directory d:/text/ เช่นกัน ให้ทำการพิมพ์คำสั่งดังนี้

```
-i d:/text/TEST_100K.txt -o d:/text/o_TEST_100K.txt
```

เมื่อโปรแกรมทำการแบ่งข้อความเสร็จ จะปรากฏข้อความ Finish