รหัสโครงการ 12p34c002

“ตัดคำไทย...ก็คล้ายๆแปลภาษาเลยนะ”

A Machine-Translation based Approach to Word Boundary Identification:
A Projective Analogy of Bilingual Translation

ประเภทโปรแกรมที่เสนอ: การแข่งขันสุดยอดซอฟต์แวร์แบ่งคำภาษาไทย (BEST 2010)

รายงานฉบับสมบูรณ์
เสนอต่อ
ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ
สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ
กระทรวงวิทยาศาสตร์และเทคโนโลยี
และ
สำนักงานส่งเสริมอุตสาหกรรมซอฟแวร์แห่งชาติ (องค์การมหาชน)

ได้รับทุนอุดหนุนโครงการวิจัย พัฒนาและวิศวกรรม
โครงการแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย ครั้งที่ 12
ประจำปีงบประมาณ 2552

โดย
ชื่อผู้พัฒนา: นายพีรเดช บางเจริญทรัพย์
ชื่ออาจารย์ที่ปรึกษา: ดร.ชัยพร ใจแก้ว

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเกษตรศาสตร์

กิตติกรรมประกาศ

ผู้พัฒนาขอขอบคุณ อาจารย์ที่ปรึกษาประจำโครงงาน ดร.ชัยพร ใจแก้ว ภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์, ดร.เทพชัย ทรัพย์นิธิ และ คุณพีรเชษฐ ปอแก้ว งานเทคโนโลยีประมวลผลข้อความ หน่วยปฏิบัติการวิจัยวิทยาการมนุษยภาษา ซึ่งให้คำปรึกษาอย่างต่อเนื่อง

โครงงาน “ ตัดคำไทย...ก็คล้ายๆแปลภาษาเลยนะ ” ได้รับทุนอุดหนุนโครงการการแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย ครั้งที่ 12 จากศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ และสำนักงานส่งเสริมอุตสาหกรรมซอฟต์แวร์แห่งชาติ

สุดท้ายนี้ ขอกราบขอบพระคุณ บิดา มารดาของข้าพเจ้าที่คอยให้ข้าพเจ้า คณะครูอาจารย์ทุกท่านที่ประสิทธิ์ประสาทความรู้แก่ข้าพเจ้า และเพื่อนๆนิสิตชั้นปีที่ 1 ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ ทุกคน

ด้วยความเคารพ

นายพีรเดช บางเจริญทรัพย์

บทคัดย่อ

คำเป็นหน่วยย่อยที่สุดที่มีความสำคัญสำหรับการประมวลผลภาษาธรรมชาติ การตัดคำไทย (Thai Word Segmentation) เป็นขั้นตอนเริ่มต้นสำหรับการประมวลผลภาษาไทย เนื่องจากภาษาไทยไม่มีสัญลักษณ์เพื่อกำกับขอบเขตของคำอย่างชัดเจน การพัฒนาระบบตัดคำภาษาไทยที่มีประสิทธิภาพจึงส่งผลโดยตรงต่อประสิทธิภาพการทำงานของระบบประมวลผลภาษาไทยในระดับสูง

รายงานฉบับนี้นำเสนอกระบวนการตัดคำที่อาศัยแนวคิดจะระบบการแปลภาษาด้วยวิธีการทางสถิติ (Statistical Machine Translation) กล่าวคือผู้พัฒนามองปัญหาการตัดคำไทยว่าสามารถมองว่าเป็นปัญหาการแปลภาษาไทย คือข้อความที่ไม่ได้รับการตัดคำ (Non-segmented Text) เป็นภาษาต้นทาง (Source Language) และข้อความที่ได้รับการตัดคำอย่างถูกต้อง (Segmented Text) เป็นข้อความผลลัพธ์หรือภาษาปลายทาง (Target Language) ระบบอาศัยข้อมูลจากแบบจำลองภาษา (Language Model) และตารางวลี (Phrase Table) ซึ่งได้จากการเรียนรู้ข้อมูลด้วยวิธีการทางสถิติ รวมถึงการอาศัยกฎการรวมคำ (Thai Character Cluster Rules) เพื่อเลือกรูปแบบการตัดคำที่ดีที่สุด

ผลการทดสอบประสิทธิภาพของระบบที่พัฒนาซึ่งเรียนรู้ด้วยคลังข้อความที่ได้รับการตัดคำแล้ว (Annotated Corpus) จำนวน 7ล้านคำ ด้วยการวัดค่า F-measure ให้ค่าเท่ากับ 94.36660% จากชุดทดสอบขนาด 1 แสนคำ

Word segmentation is a fundamental and essential tool for Thai language processing. This paper presents a framework of Statistical Phrase-based Machine Translation for Thai word segmentation. The segmentation task can be recognized as a translation process from an unsegmented sentence to a segmented sentence. We formulate the problem by mapping individual characters (Non-segmented Text) to groups of characters (Segmented Text). Language model and Phrase Table which are constructed from the training data are applied in order to search for the best segmentation result. We also utilized Thai Characters Cluster Rules (TCC) in pre-process procedure and provided a post-processing system to correct segmentation errors of unknown words. The evaluation result shows the accuracy with average F-measure of 94.36660% from 100,000 words test case.

Keywords: Word Segmentation, Language Model, Phrase Table, Thai Characters Cluster Rules

บทนำ

ปัจจุบันระบบสารสนเทศมีความต้องการนำ ระบบเพื่อจัดการงานที่เกี่ยวข้องกับข้อความและภาษาโดยอัตโนมัติมากยิ่งขึ้น เช่น ระบบการแปลภาษา, ระบบสืบค้นข้อมูล และ ระบบการสรุปความ เป็นต้น เทคโนโลยีประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) จึงเข้ามามีบทบาทเป็นอย่างมากในปัจจุบัน

ภาษาในภูมิภาคเอเชียหลายๆภาษา เช่น ภาษาไทย, ภาษาญี่ปุ่น และ ภาษาจีน จำเป็นต้องนำระบบการตัดคำมาประมวลผลข้อความขั้นต้น (Pre-processing) เนื่องจากภาษาไทยขาดสัญลักษณ์ที่แสดงขอบเขตของคำอย่างชัดเจน (Word Boundary) และภาษาไทยยังขาดตัวบ่งบอกขอบเขตในระดับพยางค์ (Syllable Boundary) ทำให้การตัดคำภาษาไทยมีความซับซ้อนมากยิ่งขึ้น อีกทั้งภาษาไทยไม่มีการกำหนดสัญลักษณ์พิเศษที่บอกถึงศัพท์ที่เป็นคำยืมจากภาษาต่างประเทศหรือชื่อเฉพาะ เช่น ภาษาญี่ปุ่นใช้อักษรคาตาคะนะ (Katakana: **かたかな**) ในการเขียนคำยืมจากภาษาต่างประเทศ และการใช้ตัวพิมพ์ใหญ่สำหรับเขียนชื่อเฉพาะเหมือนในภาษาอังกฤษ ดังนั้นการตัดคำจึงเป็นงานพื้นฐานที่สำคัญมากสำหรับการประมวลผลข้อความภาษาไทย ประสิทธิภาพของการตัดคำมีผลกระทบโดยตรงต่อความสามารถโดยรวมของระบบประมวลผลภาษาไทยระดับสูง งานวิจัยที่ผ่านมาได้นำเสนอวิธีการมากมายสำหรับการตัดคำไทย อาทิการใช้พจนานุกรม, การใช้ข้อมูลทางสถิติ และการใช้เทคนิคการเรียนรู้ด้วยเครื่อง เช่น Markov Model [4], Support Vector Machine และ Conditional Random Field [11]

จากปัญหาข้างต้น โครงงานชิ้นนี้จึงนำเสนอวิธีการใหม่สำหรับการตัดคำโดยการประยุกต์กฎการรวมอักษรภาษาไทย (TCC rules) ในการประมวลผลขั้นต้นซึ่งช่วยจัดการคำที่ไม่ปรากฏในคลังความรู้ของระบบ (Unknown Word) และลดความกำกวมของการแบ่งคำในประโยค รวมทั้งอาศัยแบบจำลองภาษาระดับ Nหน่วยคำ (N-gram Language Model) และตารางวลี (Phrase Table) โดยมองปัญหาการตัดคำเป็นรูปแบบปัญหาการแปลภาษา นั่นหมายถึงการประยุกต์ใช้วิธีการทางด้านการแปลภาษาในการตัดคำไทย โดยรายงานฉบับนี้ประยุกต์วิธีการแปลภาษาด้วยวิธีการทางสถิติ (Statistical Machine Translation) เนื่องจากเป็นกระบวนการที่ได้รับความนิยมอย่างแพร่หลายในปัจจุบัน [1]

สารบัญ

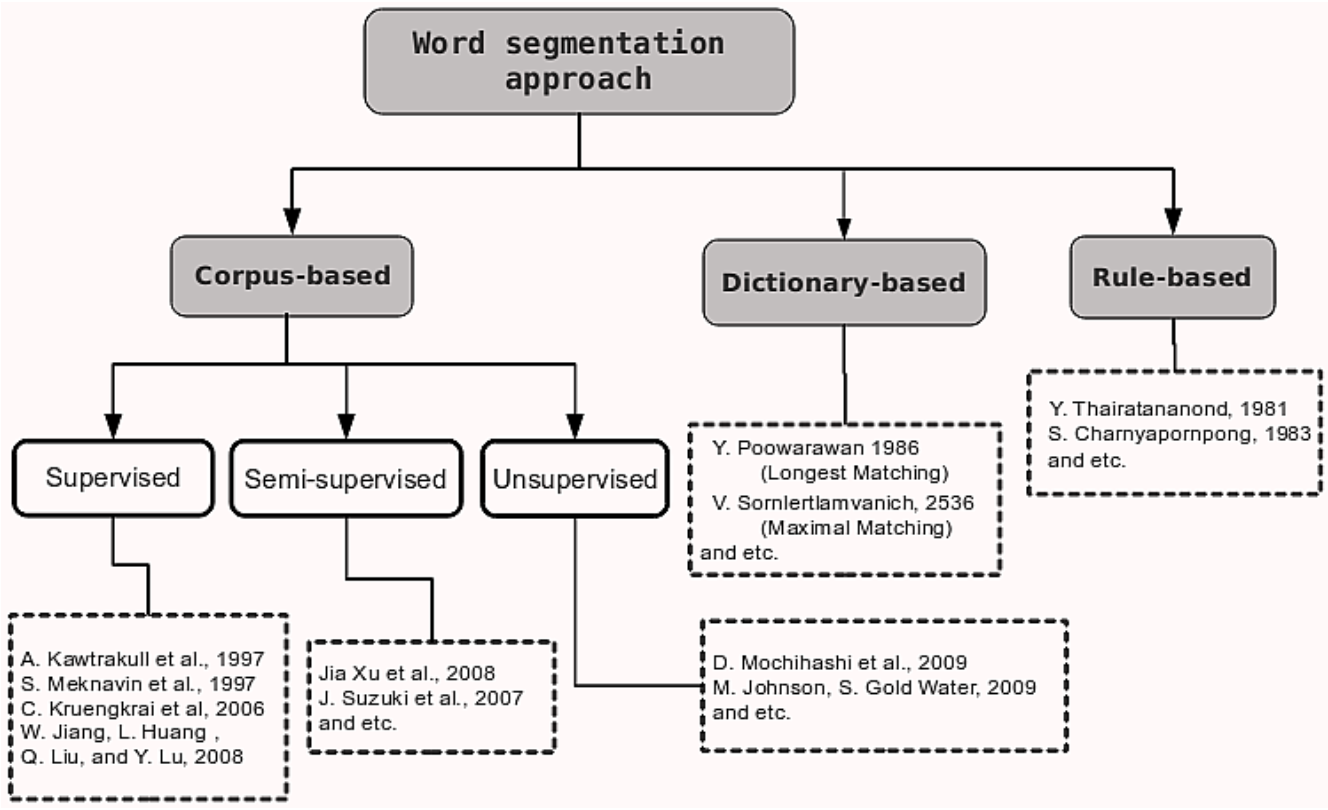
วัตถุประสงค์และเป้าหมาย

1. เพื่อพัฒนาขั้นตอนวิธี (Algorithm) ที่มีประสิทธิภาพสูงสุด (F-measure) และความเร็วในการประมวล ผลไม่ต่ำกว่า 500 คำ/วินาที
2. เพื่อออกแบบขั้นตอนวิธีการตัดคำไทยโดยประยุกต์วิธีการด้าน Statistical Machine Translation
3 เพื่อทดสอบประสิทธิภาพการประยุกต์วิธีการด้าน Statistical Machine Translation เพื่อการตัดคำภาษาไทย
4. เพื่อฝึกฝนทักษะกระบวนการทำโครงงานด้านการประมลผลภาษาธรรมชาติ (Natural Language Processing) สำหรับผู้พัฒนา
5. พัฒนาระบบตัดคำภาษาไทยที่สามารถจัดการกับคำที่ไม่รู้จักและคำที่กำกวมอย่างเหมาะสม

รายละเอียดของโปรแกรมที่พัฒนาขึ้น

งานวิจัยที่เกี่ยวข้อง

เนื่องจากระบบตัดคำเป็นเครื่องมือขั้นพื้นฐานที่จำเป็นอย่างยิ่งสำหรับงานวิจัยด้านการประมวลผลภาษาธรรมชาติสำหรับภาษาไทย งานวิจัยในด้านนี้ได้เริ่มมาเป็นระยะเวลากว่า 30 ปี ซึ่งในแต่ละงานวิจัยต่างนำเสนอวิธีการที่แตกต่างกัน ซึ่งสามารถแบ่งประเภทดังแผนภาพต่อไปนี้



ภาพที่ 1. ประเภทของวิธีการสำหรับตัดคำในปัจจุบัน

โดยรายละเอียดอย่างคร่าวๆ ของวิธีการในประเภทต่างเป็นดังนี้

- วิธีการตัดคำด้วยกฎทางภาษา (Rule-based): วิธีการนี้เป็นวิธีที่ใช้ในยุคแรก เนื่องจากในยุคดังกล่าว คอมพิวเตอร์ยังมีประสิทธิภาพค่อนข้างต่ำ ประกอบกับหน่วยความจำที่มีไม่มากนัก ดังนั้นในยุคแรกจึงมีการใช้กฎทางไวยกรณ์ภาษาเพื่อแบ่งพยางค์ของคำ

•วิธีการตัดคำด้วยพจนานุกรม (Dictionary-based): เนื่องจากการใช้กฎเพียงอย่างเดียวไม่สามารถตัดคำได้อย่างแม่นย ำ ประกอบกับคอมพิวเตอร์มีประสิทธิภาพสูงมากขึ้น พจนานุกรมจึงถูกนำมาใช้ประกอบในการตัดค ำ แต่งานวิจัยส่วนใหญ่มักจะนำกฎมาประกอบ เพื่อแก้ไขปัญหาคำที่ไม่ปรากฏในพจนานุกรม

•วิธีการตัดคำด้วยคลังข้อความ (Corpus-based): ระบบการตัดคำประเภทนี้จะอาศัยคลังประโยค (Corpus) มาช่วยเป็นข้อมูลในการตัดค ำ โดยการประยุคหลักการทางสถิติและการเรียนรู้ด้วยเครื่อง (Machine Learning) ซึ่งงานวิจัยประเภทนี้ยังแบ่งได้เป็น 3 ประเภทดังนี้

° Supervised Approach: ข้อมูลที่ใช้ในการเรียนรู้ (Train) ระบบ ต้องเป็นข้อมูลที่ได้รับการกำกับขอบเขตของคำ (Annotated Corpus) ด้วยนักภาษาศาสตร์แล้ว

° Semi-supervised Approach: ข้อมูลสำหรับเรียนรู้ระบบ จะเป็นทั้งข้อมูลที่ได้รับการกำกับ และไม่ได้กำกับขอบเขตคำ โดยส่วนใหญ่ข้อมูลที่กำกับแล้วจะมีจำนวนน้อย

° Unsupervised Approach: ระบบประเภทนี้สามารถเรียนรู้ จากคลังข้อความที่ไม่ได้กำกับจากนักภาษาศาสตร์ แต่ในปัจจุบันวิธีการประเภทนี้ยังมีประสิทธิภาพต่ำกว่า วิธีการสองประเภทแรก

ส่วนต่อไปจะแสดงรายละเอียดของงานวิจัยต่างๆ โดยรายงานชิ้นนี้จะยกตัวอย่างวิธีการที่น่าสนใจมาแสดงรายละเอียดดังต่อไปนี้

•***การตัดคำแบบเลือกคำยาวที่สุด*** *(Longest Matching)*

วิธีการ Longest Matching (ยืน ภู่วรวรรณ และ วิวรรธน์ อิ่มอารมณ์, 2529) เป็น Algorithm แรกที่มีการนำพจนานุกรมคำศัพท์มาใช้ในการตัดคำ นอกจากนี้มีการนำกฎทางไวยกรณ์จำนวน 18 กฎมาใช้ด้วยในกรณีที่พบ Unknown Word

วิธีการนี้จะตรวจสอบตัวอักษรจากซ้ายไปขวา โดยนำไปตรวจกับพจนานุกรมหากไม่พบคำดังกล่าวก็จะตัดอักษรตัวขวาสุด แล้วทำตามขั้นตอนเดิมไปเรื่อยๆ ซึ่งวิธีการแสดงดังภาพที่ 1. เมื่อคำที่พิจารณาคือ “แนวทางการแก้ปัญหาหนี้สินเกษตร” [7]

ลำดับการทำงาน	ส่วนของคำที่ยาวที่สุด	ส่วนที่เหลือ
1	แนวทาง	การแก้ปัญหาหนี้สินเกษตรกร
2	การ	แก้ปัญหาหนี้สินเกษตรกร
3	แก้	ปัญหาหนี้สินเกษตรกร
4	ปัญหา	หนี้สินเกษตรกร
5	หนี้สิน	เกษตรกร
6	เกษตรกร	

ภาพที่ 2. ลำดับการทำงานของ Longest Matching

นอกจากนี้ในกรณี เมื่อเลือกคำยาวที่สุดแล้ว ทำให้ในคำถัดไปเกิดพยางค์ที่ไม่เกิดความหมาย จะสามารถย้อนกลับแล้วเลือกคำที่มีความยาวลองลงมาได้ (Back Tracking) ซึ่งแสดงดังตัวอย่างต่อไปนี้ เมื่อคำที่พิจารณาคือ “โคนมนอนบนกองหญ้า” [8]

ลำดับ	ประโยค	คำที่ได้	คำที่เลือก
1	โคนมนอนบนกองหญ้า	โคน, โค	โคน
2	มนอนบนกองหญ้า	-	<ย้อนกลับ>
3	โคนมนอนบนกองหญ้า	โคน, โค	โค <คำรอง>
4	นมนอนบนกองหญ้า	นม	นม
5	นอนบนกองหญ้า	นอ, นอน	นอน
6	บนกองหญ้า	บน	บน
7	กองหญ้า	กอ, กอง	กอง
8	หญ้า	หญ้า	หญ้า

ภาพที่ 3. ลำดับการย้อนกลับของ Longest Matching

•***การตัดคำโดยเลือกแบบเหมือนมากที่สุด*** *(Maximal Matching)*

การตัดคำแบบ Maximal Matching (วิรัช ศรเลิศล้ำวาณิช, 2536) เป็นวิธีการที่กระทำบนผลลัพธ์ที่ได้จากวิธีการ Longest Matching เพื่อเป็นการแก้ปัญหาการเลือกตัดคำที่ยาวเกินไปตั้งแต่ครั้งแรกเช่น ประโยค “ไปหามเหสี” ซึ่งหากนำไปตัดด้วยวิธีการ Longest Matching จะให้ผลลัพธ์เป็น “ไป|หาม|เห|สี”

โดย Maximal Matching จะหาวิธีการตัดคำที่เป็นไปได้ทั้งหมด โดยการย้อนกลับเข้าไปประมวลผลด้วยวิธีการ Longest Matching ในคำที่ได้จากการตัดคำด้วย Longest Matching ครั้งแรก ซึ่งเรียกวิธีการนี้ว่า “Backtracking” เช่น ประโยค “ไป|หาม|เหสี” ในคำว่า “หาม” จะถูกแยกเป็น “หาม” และ “หา” แล้วนำ “ม” ไปรวมกับคำถัดไป ดังนั้นทางเลือกทั้งหมดคือ “ไป|หา|มเหสี” และ “ไป|หาม|เห|สี” ขั้นตอนต่อไปคือเลือกทางเลือกที่มีจำนวนคำน้อยที่สุด ซึ่งในที่นี้คือ “ไป|หา|มเหสี” แต่หากจำนวนคำเท่ากันก็จะย้อนกับไปใช้กฎเกณฑ์ของ Longest Matching อีกครั้งคือเลือกคำที่ยาวที่สุด เช่น ประโยค “ฉันนั่งตากลม” ซึ่งสามารถตัดได้เป็น “ฉัน|นั่ง|ตา|กลม” และ “ฉัน|นั่ง|ตาก|ลม” แต่เมื่อพิจารณาแล้วจะเลือก “ฉัน|นั่ง|ตาก|ลม”

•***การตัดคำด้วยการประยุกต์แบบจำลองไตรแกรม*** *(Trigram Model)*

ในงานวิจัยเรื่อง A Statistical Approach for Thai Morphological Analyzer [4] ได้นำเสนอการประยุกต์หลักการ Tri-gram Model [9] เพื่อคำนวณความน่าจะเป็นของกลุ่มคำ (Word cluster) เช่น คำที่อยู่ด้านหน้าสองคำจะมีผลอย่างไรกับคำในปัจจุบัน ซึ่งอธิบายดังสมการ (1)

$$P(w_{t,n}) = \prod_{i=1}^{n} P(w_i \mid w_{i-2,i-1}) \qquad (1)$$

การคำนวณค่าความน่าจะเป็นตามสมการนั้นจะต้องใช้คลังข้อความขนาดใหญ่มาก โดยคลังข้อความควรจะมากกว่า N^3 คำ โดยที่ N คือจำนวนคำที่เป็นไปได้ทั้งหมด เนื่องจากวิธีนี้ต้องมีการนำค่าสถิติของการเกิดรวมกันของคำ 3 คำที่เรียงติดกัน มาใช้ในการคำนวณ ดังนั้นการใช้แบบจำลอง Tri-gram ต้องการคลังข้อมูลขนาดใหญ่ ในกรณีที่ไม่สามารถเตรียมคลังข้อความได้เพียงพอ จะทำให้ความถี่ของการเรียงในรูปแบบ Tri-gram มีน้อยจึงเกิดปัญหาที่เรียกว่า Sparse-data problem [4] เพื่อแก้

ปัญหาดังกล่าวจึงใช้สมการ (2) แทน

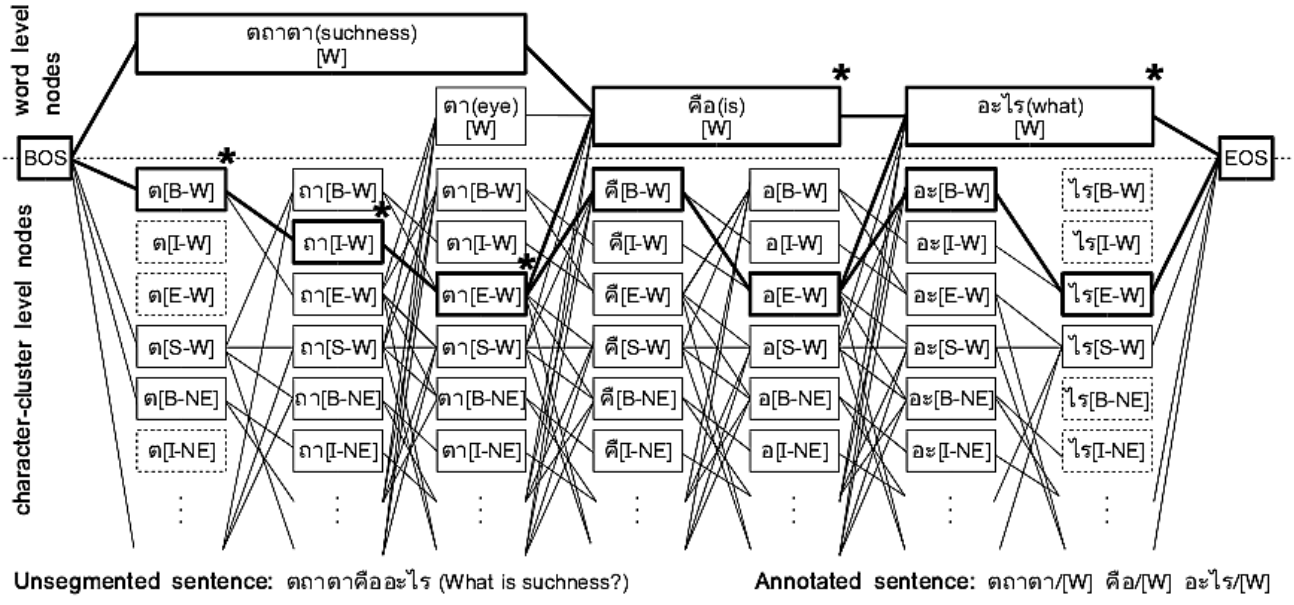
$$P(w_i \mid w_{i-2,i-1}) = \prod_{i=1}^{n} \{\lambda_1 P(w_n) + \lambda_2 P(w_n \mid W_{n-1}) + \lambda_3 P(w_n \mid w_{n-2,n-1})\} \quad (2)$$

จากสมการข้างต้น ระบบจะสามารถคำนวณความน่าจะเป็นได้ดีขึ้น แม้ขาดการเรียงตัวของคำรูปแบบ Tri-gram และ Bi-gram ในคลังข้อความ (Corpus) โดยจากการทดลอง ค่าที่ดีที่สุดในงานด้านภาษาไทยของ คือ 0.1, 0.3 และ 0.6 ตามลำดับ [4]

- ***A Word and Character-Cluster Hybrid Model for Thai Word Segmentation***

งานวิจัยฉบับนี้ [12] เป็นการนำเสนอวิธีการเพื่อแก้ไขปัญหาคำที่ไม่ปรากฏในคลังความรู้ของระบบ (Unknown Word) ซึ่งผู้พัฒนาได้นำเสนอ แบบจำลองผสมระหว่างคำและกลุ่มอักขระ ซึ่ง " กลุ่มอักขระ " หมายถึง การนำอักขระในคำที่สนใจไปผ่านการรวมกลุ่มด้วยกฎทางภาษา โดยงานวิจัยชิ้นนี้มีการใช้กฎจำนวน 16 กฎ

กระบวนการที่นำเสนอในงานวิจัยชิ้นนี้ จะสร้างกราฟทางเลือก (Lattice) ของคำที่นำเข้ามา โดยกราฟจะประกอบด้วย สถานีเชื่อมโยง (Node) ในระดับคำและกลุ่มอักขระ ดังแผนภาพ



ภาพที่ 4. กราฟทางเลือกที่ถูกสร้างขึ้นเมื่อข้อความนำเข้าคือ ตถาตาคืออะไร [12]

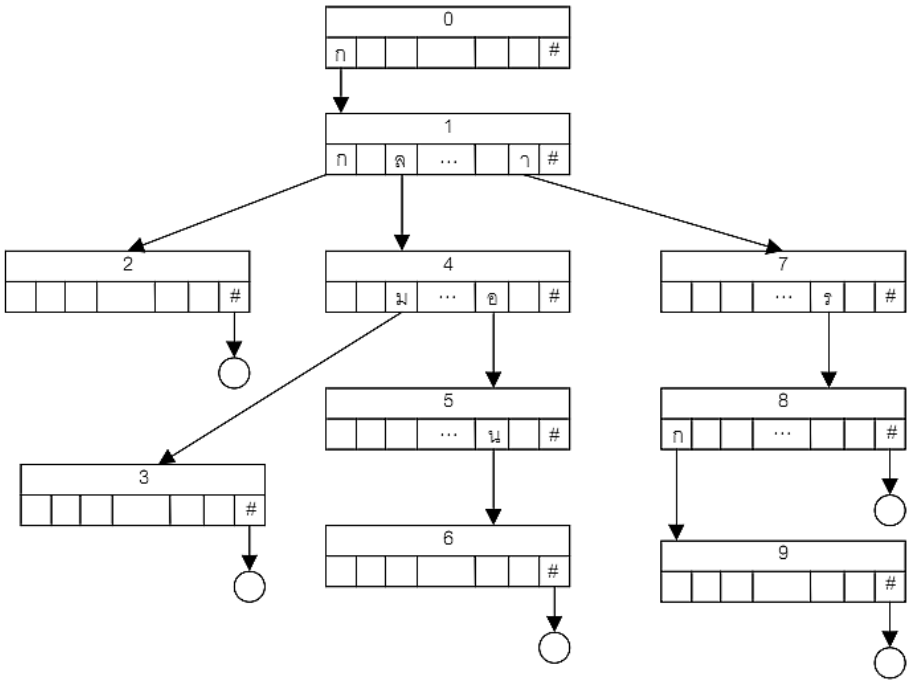
ในขั้นตอนการเรียนรู้ระบบ (Traing Phase) ข้อความจะถูกสร้างเป็น กราฟทางเลือกดังภาพที่ 4 โดยเส้นเชื่อม (Edge) ที่เป็นเส้นหนาคือ เส้นทางที่ถูกกำกับตามคลังข้อความที่ถูกเรียนรู้ และทางเลือกที่ถูกกำกับด้วยเครื่องหมายดอกจัน (*) คือเส้นทางที่ถูกเลือกเพื่อให้ระบบเรียนรู้ จะเห็นว่ารายงานชิ้นนี้ไม่ได้เลือก เส้นทางที่ถูกกำกับตามคลังข้อความ ในช่วงคำว่า "ตถาตา” เนื่องจากคำดังกล่าวถูกนิยามว่าเป็น Unknown Word (กำหนดจากจำนวนครั้งที่ปรากฏในคลังข้อความ) เพื่อให้ระบบได้เรียนรู้ลำดับของ Unknown Word ประโยชน์ที่ได้รับคือ ระบบจะมีความสามารถในการจัดการกับคำที่ไม่ปรากฏในคลังข้อความได้อย่างมีประสิทธิภาพมากขึ้น

จากผลการทดสอบประสิทธิภาพ ด้วยข้อความทดสอบจาก InterBEST 2009 [13] ทั้งหมด 12 ชุด ค่า F-measure เฉลี่ยเท่ากับ 96.853% โดยความแม่นยำสูงสุดเท่ากับ 98.524% ในชุดทดสอบหมวดพุทธศาสนา และค่า F-measure ต่ำสุด 92.299% ในหมวดข่าวในพระราชสำนัก (Out of Training Data)

ทฤษฎีหลักการและเทคนิคที่ใช้

•โครงสร้างข้อมูลแบบทรี Trie

คำว่า Trie ย่อมาจากคำว่า "Retrieval” ดังนั้นจึงออกเสียงว่า " ทรี /triː/ " (Edward Fredkin, 1960) โครงสร้างข้อมูลประเภทนี้มีลักษณะคล้ายโครงสร้างแบบต้นไม้ แต่มีลักษณะการเก็บข้อมูลที่ต่างกัน โดยมีรูปแบบการเก็บข้อมูลดังภาพที่ 5. ซึ่งในงานวิจัยส่วนใหญ่นิยมใช้ชนิดข้อมูลแบบนี้ในการเก็บพจนานุกรม เนื่องจากสามารถเข้าถึงข้อมูลได้อย่างรวดเร็ว ซึ่งสามารถเข้าถึงข้อมูลเพียง $O(n)$ ครั้ง ในงานวิจัยชิ้นนี้ใช้ชนิดทรี ในการเก็บคลังคำศัพท์และความถี่ที่ปรากฏในคลังข้อความ



ภาพที่ 5. โครงสร้างข้อมูลแบบทรี [8]

จากภาพตัวอย่าง หากต้องการสืบค้นคำว่า "กลม” โปรแกรมจะเข้ามาใน Node แรกซึ่งเก็บอักษรนำของคำทุกคำในพจนานุกรม และข้ามไปยัง Node หมายเลข 1 ซึ่งมีอักษร 'ล' เป็นสมาชิกอยู่ ผ่านไป Node ที่ 4 และสุดม้ายไปยัง Node หมายเลข 3 ซึ่งมีเครื่องหมายจบของคำ (#) เป็นสมาชิกหนึ่ง นั่นแปลว่ามีคำที่ประกอบด้วย อักษร 'ก' 'ล' 'ม' และจุดจบของคำ (#) ตามลำดับ

•วิธีการตัดคำภาษาไทยที่นำเสนอ (Proposed Algorithm)

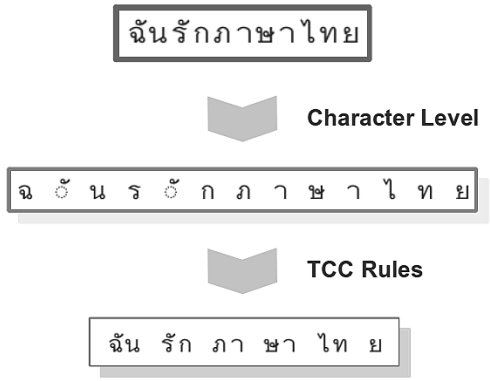
กระบวนการตัดคำภาษาไทยที่นำเสนอประกอบด้วย 3 ส่วนหลัก คือ ขั้นตอนเตรียมข้อมูล (Pre-processing), ขั้นตอนระบุขอบเขตของคำ (Segmentation) และ ขั้นตอนประมวลผลหลังการตัดคำ (Post-processing) ซึ่งรายละเอียดดังนี้

I.ขั้นตอนเตรียมข้อมูล (Pre-processing)

1) แปลงข้อมูลให้อยู่ในระดับอักษร (Character Level): งานวิจัยชิ้นนี้มองมองปัญหาการตัดคำว่า เป็นการรวมกลุ่มลำดับของอักษร ให้กลายเป็นคำที่ถูกต้อง ดังนั้นข้อความนำเข้าจึงจำเป็นจะต้องถูกแปลงเป็นข้อมูลระดับอักษร

2) รวมกลุ่มอักษรด้วยกฎการรวมอักษรไทย (Thai Character Clustering Rules: TCC) [9]:

ขั้นตอนนี้จะรวมลำดับของอักษรให้เป็นกลุ่มอักษร ซึ่งไม่สามารถแยกย่อยได้อีกตามหลักไวยกรณ์ภาษา ดังภาพ 6. กฎการรวมอักษรมาจากลักษณะของภาษาไทยที่อักขระบางตัวสามารถวางในตำแหน่งที่จำกัด ในงานชิ้นนี้ใช้กฎจำนวน 54 กฎ



ภาพ 6. กระบวนการเตรียมข้อมูล (Pre-processing)

II.ขั้นตอนระบุขอบเขตของคำ (Segmentation)

1) สร้างทางเลือกที่น่าจะเป็นไป: ในขั้นตอนแรกระบบจะสร้างทางเลือกในการตัดคำที่น่าจะเป็นทั้งหมด โดยข้อมูลจากตารางวลี (Phrase Table) โดยงานวิจัยชิ้นนี้นำเสนอวิธีการตัดคำที่มองในระดับของกลุ่มคำ (Phrase) โดยที่กลุ่มคำไม่จำเป็นต้องมีความทางภาษา ซึ่งรายงานชิ้นนี้กำหนดให้ขนาดกลุ่มคำไม่เกิน 4 คำ การสร้างทางเลือกของคำโดยมองในระดับกลุ่มคำจะทำให้ข้อมูลของคำบริบทถูกใช้ประโยชน์

2) ตัดสินใจเลือกรูปแบบการตัดคำ: ในขั้นตอนแรกระบบได้สร้างทางเลือกที่น่าจะเป็นทั้งหมด ในขั้นตอนระบบจะตัดสินเลือกรูปแบบที่เป็น รูปแบบการตัดคำที่เหมาะสมที่สุด โดยประโยคที่ถูกเลือกจะเป็นประโยคที่ให้ค่าความน่าจะเป็นสูงสุด ดังสมการ (3)

$$T_{best} = argmax_T \{ \prod_{j=1}^{K} P(\varphi_j^T | tcc_{j1}^T, tcc_{j2}^T \ldots tcc_{jm}^T) \times \prod_{i=6}^{L} P(w_i^t | w_{i-1}^t, w_{i-2}^t, \ldots, w_{i-6}^t) \times \omega^{length(T)} \} \quad (3)$$

เมื่อ φ_j^T แทน กลุ่มคำลำดับที่ j ของประโยคทางเลือก T

w_i^t แทน คำลำดับที่ i ของประโยคทางเลือก T

tcc_{jk}^T แทน กลุ่มอักษรลำดับที่ k ของกลุ่มคำ φ_j^T

$\omega^{length(T)}$ แทน ค่าคงที่ยกกำลังด้วยจำนวน Node ของกราฟ

โดยที่ K หมายถึงจำนวนกลุ่มคำ (φ) และ L หมายถึงจำนวนคำในประโยค สำหรับการคำนวณค่าความน่าจะเป็นเพื่อใช้คำนวณดัง สมการที่ 3 ได้จากการคำนวณข้อมูลจากคลังข้อความของ BEST 2010

$P(\varphi_j^T | tcc_{j1}^T, tcc_{j2}^T \ldots tcc_{jm}^T)$

$\omega^{length(T)}$

ขนาด 7 ล้านคำ โดยใช้เครื่องมือสำหรับสร้างแบบจำลองภาษา SRILM [14] และ คำนวณโดยขึ้นอยู่กับความถี่ในการเกิดลำดับของกลุ่มอักษร (*tcc*) ดังกล่าวใน คลังข้อความ และ พจน์ของ จะทำให้ระบบเลือกเส้นทางที่มี Node น้อยกว่า

III.ขั้นตอนประมวลผลหลังการตัดคำ (Post-processing)

1) แก้ไขความผิดพลาดเนื่องจากสัญลักษณ์ทัณฑฆาต: ในคำบางคำโดยเฉพาะคำที่ไม่ปรากฏในคลังข้อความ (Unknown Word) ซึ่งส่วนใหญ่เป็นคำที่ยืมจากภาษาต่างประเทศ ดังนั้นมักจะมีเครื่องหมายทัณฑฆาต งานชิ้นนี้จึงออกแบบระบบเพื่อตรวจสอบตำแหน่งของทัณฑฆาต ในประโยคที่ถูกเลือกจากขั้นตอน 2.2 เช่น เทอ | ***ร์*** | โม | ได | นา | มิก | ***ส์*** | เป็นต้น

2) แก้ไขความผิดพลาดเนื่องจากอักษรเดี่ยว: อักษรในภาษาไทยทุกตัว ยกเว้น ณ และ ธ ไม่สามารถแยกออกจากอักขระอื่นๆได้ จากเหตุผลดังกล่าวงานวิจัยชิ้นนี้จึงได้สร้างโปรแกรมเพื่อตรวจสอบและแก้ไขอักษรที่ถูกตัดลำพัง เช่น เขา | เป็น | ***อ / ส / ร*** | พิษ เป็นต้น

เครื่องมือที่ใช้ในการพัฒนา

1. Python Version 2.6.4
2. Geany IDE 0.18
3. Moses 2008-7-11: Library รวมคำสั่งด้านการแปลภาษาด้วยวิธีการทางสถิติ [1]
4. Diffuse 0.3.3: โปรแกรมเพื่อแสดงความแตกต่างของไฟล์เอกสาร

รายละเอียดโปรแกรมที่พัฒนาในเชิงเทคนิค

- *Input / Output Specification*

° Input: ข้อมูลนำเข้าเป็นไฟล์ข้อความ (Text File) ที่ถูกเข้ารหัสแบบ UTF-8

° Output: ข้อมูลออกเป็นไฟล์ข้อความ (Text File) ที่ถูกเข้ารหัสแบบ UTF-8 และถูกแบ่งขอบเขตของคำด้วยเครื่องหมาย Pipe (“ | “)

- *Functional Specification*

° ผู้ใช้สามารถแบ่งคำหลายไฟล์พร้อมกันได้

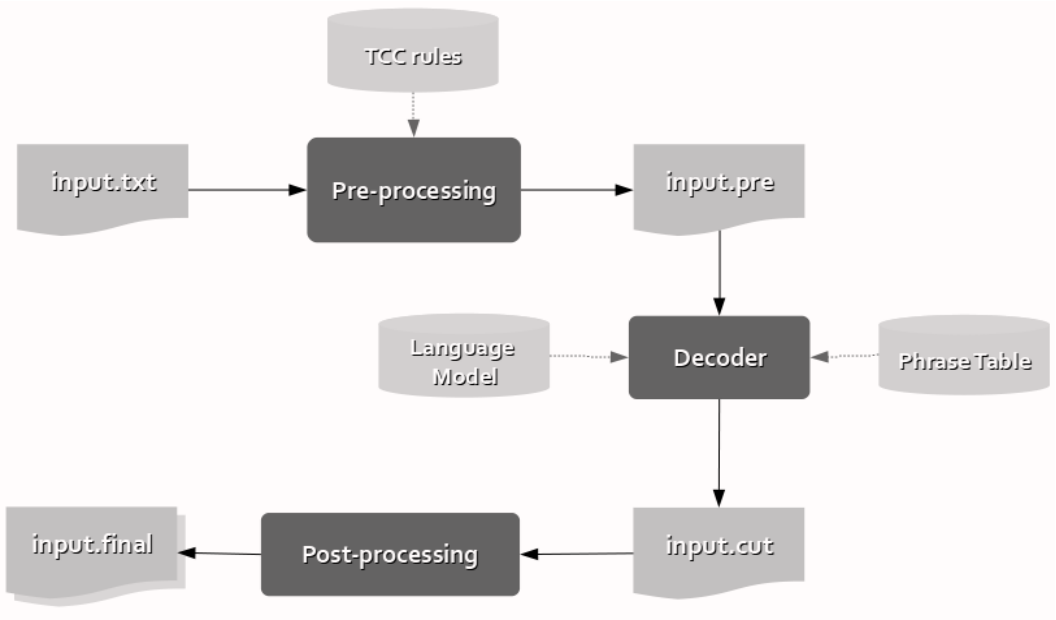
•*โครงสร้างของซอฟต์แวร์*

ขั้นตอนการทำงานแบ่งออกเป็นสามส่วนหลัก คือ

1.Pre-processing: ข้อมูลที่นำเข้าสู่ระบบจะประมวลผลด้วยกฎการรวมคำไทย (TCC rules) และแปลงข้อมูลให้อยู่ในระดับอักษร (Character Level)

2.Decoder: ข้อมูลจากชุดแรกถูกคำนวณและประมวลผลโดยวิธีการที่นำเสนอ โดยพยายามหาทางเลือกในการตัดคำที่ดีที่สุด โดยอาศัยข้อมูลทางสถิติจาก Language Model และ Phrase Table

3.Post-processing: เนื่องจากคำบางคำอาจเป็นคำที่ไม่ได้อยู่ในคลังความรู้ของระบบ (Unknow Word) ทำให้เกิดผลลัพธ์ที่ผิดพลาด ผู้พัฒนาจึงองค์ความรู้ทางไวยกรณ์ภาษาไทยและข้อมูลจากการวิเคราะห์คลังข้อความ (Corpus) มาเพิ่มความถูกต้องมากยิ่งขึ้น



ภาพที่ 3. โครงสร้างการทำงานของซอฟต์แวร์

ขอบเขตและข้อจำกัดของโปรแกรม

•โปรแกรมที่พัฒนาขึ้นสามารถใช้งานได้ดีกับข้อความภาษาไทย

•ระบบที่พัฒนาเป็นระบบแบบ Supervised Approach จึงจำเป็นต้องฝึกฝน ระบบด้วยคลังข้อความภาษาไทยที่ผ่านการกำกับขอบเขตคำ อย่างถูกต้อง แล้วเท่านั้น (Annotated Corpus)

•โปรแกรมที่พัฒนารองรับข้อมูลที่ถูกเข้ารหัสแบบ UTF-8 เท่านั้น

•ความเร็วในการประมวลผลไม่ต่ำกว่า 500 คำ/วินาที

•โปรแกรมแสดงขอบเขตของคำด้วยเครื่องหมาย |

กลุ่มผู้ใช้โปรแกรม

•ผู้พัฒนาระบบด้านการประมวลผลภาษาธรรมชาติที่ต้องการ ระบบตัดคำไทย

•ผู้สนใจทั่วไป

ผลการทดสอบโปรแกรม

จากการทดสอบประสิทธิภาพของระบบที่นำเสนอ โดยใช้คลังข้อความขนาด 7 ล้านคำ จาก BEST 2010 ซึ่งประกอบด้วยข้อความ 8 ประเภทดังนี้

- Article
- Buddhism
- Encyclopedia
- Law
- News
- Novel
- Talk
- Wiki

เมื่อทดสอบโปรแกรมด้วยข้อความทดสอบ (Test Case) ขนาด 1 แสนคำ ได้ผลการทดลองดังตาราง

	Precision	Recall	F-measure
BEST 2009	96.51164%	97.50746%	97.00700%
BEST 2010	92.69922%	96.09506%	**94.36660%**

ตารางที่ 1. ผลการทดสอบด้วยข้อมูลหนึ่งแสนคำ

ปัญหาและอุปสรรค

1.คลังข้อความ (Annotated Corpus) มีการแบ่งคำที่ผิดจากหลักทางภาษา จึงจำเป็นต้องทำความสะอาดข้อมูลก่อน เช่น " ประ|ชาม|ติ ” และ " จาน|แม่เ|หล็ก|ชนิด|แข็ง "

2.ความกำกวมในการระบุขอบเขตของคำในคลังข้อความ (Inconsistency) หมายถึง วลีที่มีบริบทใกล้เคียงกันแต่ได้รับการระบุขอบเขตของคำแตกต่างกัน เช่น " อยู่|ใต้|ร่มเงา|ของ " แต่บางวลีในคลังข้อความพบว่าถูกกำกับเป็น " อยู่|ใต้|ร่ม|เงา|ของ " เป็นต้น

3.โปรแกรมจำเป็นต้องใช้เวลาในการนำเข้าคลังความรู้เข้าสู่ระบบ (Load Model) ผู้พัฒนาจึงแก้ไขด้วยการแปลงตารางวลีให้อยู่ในรูปแบบ Binary Phrase Table และเลือก Load ตามความต้องการ (On-demand Loading)

แนวทางในการพัฒนาและประยุกต์ใช้งาน

1.ระบบตัดคำไทยที่พัฒนาสามารถใช้ร่วมกับ โปรแกรมประมวลผลภาษาไทยต่างได้ เช่น ระบบแปลภาษา (Machine Translation), ระบบ Question and Answer และอื่นๆ

2.พัฒนาโปรแกรมเป็นรูปแบบ Object Oriented เพื่อให้สะดวกต่อผู้ทีต้องการพัฒนาต่อยอดหรือประยุกต์ใช้งาน

3.เพิ่มระบบ Name Entity Detection เพื่อกำหนดขอบเขตของนิพจน์ระบุนาม(Name Entity) ก่อนประมวลผล

เอกสารอ้างอิง

[1]Koehn, P., et al, “Moses: Open source toolkit for statistical machine translation,”. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, demonstation session ,Prague, June 2007, pp 177–180.

[2]Stolcke, Andreas, “SRILM-An Extensible Language Modeling Toolkit,” In International Conference on Spoken Language Processing, Denver, Colorado, September 2002, pp 901-904.

[3]Khankasikam, K. and Muansuwan, N., "Thai Word Segmentation a Lexical Semantic Approach", The 10th Machine Translation Summit (MT Summit X), September 12-16 2005, Phuket, Thailand, pp. 331.

[4]Kawtrakul Asanee , Thumkanon Chalathip, “A Statistical Approach to Thai Morphological Analyzer,” Dept. of Computer Engineering, Kasetsart University, Bangkok.

[5]Robert Dale, Hermann Moisl, Harold Somers, “Handbook of Natural Language Processing,” Basel, New York: Marcel Dakker, 2000

[6]P. Pisit, Teng-Amnuay Yunyong, Performance Comparison of Thai Word Separation Algorithms. Proceedings of the National Computer Science and Engineering Conference 1998 (NCSEC'98), 19th-21st October 1998. Bangkok.

[7]S. Sudprasert 2005, “Thai Word Segmentation based on Global and Local Unsupervised Learning,” [in Thai], Master Thesis of Engineering, Department of Computer Engineering, Kasetsart University, Bangkok.

[8]C. Paisarn 1998. “Feature-based Thai Word Segmentation,” [in Thai] Master Thesis of Engineering, Department of Computer Engineering, Chulalongkorn University, Bangkok.

[9]T. Threeramunkong, V. Sornlertlamvanich, T. Tanhermhong, and W. Chinnan, “Character Cluster Based Thai Information Retrieval,” in Proceedings of EMNLP, 1996, pp. 133-142

[10]Y. Poovarawan and W. Imarrom, “Thai Syllable Separator by Dictionary,” [in Thai] Proceedings of the 9th Annaul Meeting on Electrical Engineering of the Thai Universities, Khonkaen, 1986

[11]C. Haruechaiyasak, and S. Kongyoung, “TLex: Thai Lexeme Analyser Based on the Consitional Random Fields,” Proceedings of InterBEST 2009: Thai Word Segmentation Workshop, pp. 13-17, 2009

[12]C. Kruengkrai, K. Uchimoto, and J. Kazama, “A Word and Character-Cluster Hybrid Model for Thai Word Segmentation,” Proceedings of InterBEST 2009: Thai Word Segmentation Workshop, pp. 24-29, 2009

[13]Human Language Technology Laboratory, National Electrics and Computer Technology Center, “InterBEST 2009 Thai Word Segmentation: ans International Episode,” [Online]. Available: URL: http:// thailang.nectec.or.th/interbest

[14]A. Stolcke, “SRILM-An Extensible Language Modeling Toolkit,” In International Conference on Spoken Language Processing, Denver, Colorado, September 2002, pp. 901-904

ภาคผนวก

คู่มือการติดตั้งและใช้งานอย่างละเอียด

1.ติดตั้ง Python ด้วยการเปิด Terminal แล้วพิมพ์คำสั่ง

```
sudo apt-get install python2.6
```

1.ติดตั้ง Moses โดยการเปิดไฟล์ moses_20080711-2nlp3~0hardy1_i386.deb

2.เรียกใช้โปรแกรมด้วยคำสั่ง

```
python 12P34C002.py input_file output_file
```

3.ประมวลผลหลังการตัดคำ (Post-processing) โดยผลลัพธ์การตัดคำคือ ไฟล์ *output_file*