การแปลภาษาไทยเป็นภาษาไทยล้านนา

THAI TO THAI-LANNA MACHINE TRANSLATION

ประธาน คำจินะ

PRATHAN COMEJINA

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

KMITL-2013-IT-M-001-002

การแปลภาษาไทยเป็นภาษาไทยล้านนา

THAI TO THAI-LANNA MACHINE TRANSLATION

ประธาน คำจินะ

PRATHAN COMEJINA

- Copy ถึง ถึงหน้า ๑๑๗.

ไก้

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

KMITL-2013-IT-M-001-002

THAI TO THAI-LANNA MACHINE TRANSLATION

PRATHAN COMEJINA

A THESIS SUBMITTED IN PATIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2013

KMITL-2013-IT-M-001-002

คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การแปลภาษาไทยเป็นภาษาไทยล้านนา
Thai to Thai-lanna Machine Translation

นักศึกษา นายประธาน คำจินะ

รหัสประจำตัว 51066409

ปริญญา วิทยาศาสตรมหาบัณฑิต

สาขาวิชา เทคโนโลยีสารสนเทศ

อาจารย์ที่ปรึกษาวิทยานิพนธ์ รองศาสตราจารย์ ดร.พรฤดี เนติโสภากุล

คณะกรรมการสอบวิทยานิพนธ์	ลายมือชื่อ
รองศาสตราจารย์ ดร.วรพจน์ กรีสุระเดช	
รองศาสตราจารย์ ดร.นุชรี เปรมชัยสวัสดิ์	
รองศาสตราจารย์ ดร.อาริต ธรรมโน	
รองศาสตราจารย์ ดร.พรฤดี เนติโสภากุล	
ผู้ช่วยศาสตราจารย์ ดร.กันต์พงษ์ วรรัตน์ปัญญา	

วัน / เดือน/ ปี ที่สอบ วันอังคารที่ 9 เมษายน 2556 เวลา 09.30 น.

สถานที่สอบ ณ ห้อง 335 (ชั้น 3) คณะเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศรับรองแล้ว

(รองศาสตราจารย์ ดร.จันทร์บูรณ์ สถิตวิริยวงศ์)
คณบดีคณะเทคโนโลยีสารสนเทศ
วันที่...14...**เดือน**...พฤษภาคม...**พ.ศ.**...2556...

ข้อวิทยานิพนธ์	การแปลภาษาไทยเป็นภาษาไทยล้านนา
นักศึกษา	นายประธาน คำจินะ
รหัสนักศึกษา	51066409
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยการสารสนเทศ
พ.ศ.	2556
อาจารย์ที่ปรึกษา	รศ.ดร.พรฤดี เนติโสภากุล

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้นำเสนอกระบวนการแปลภาษาไทยเป็นภาษาไทยล้านนาในระดับประโยค มีกระบวนการประมวลผลอยู่ 3 ขั้นตอนหลักได้แก่ ขั้นตอนก่อนการประมวลผล ขั้นตอนการแปลประโยคไทยล้านนาและขั้นตอนการสร้างประโยคไทยล้านนา ในขั้นตอนก่อนการประมวลผลจะประกอบด้วยขั้นตอนย่อย 2 ขั้นตอนคือ การตัดคำในงานวิจัยนี้ใช้เครื่องมือ KU-CUT ของมหาวิทยาลัยเกษตรศาสตร์และขั้นตอนวิเคราะห์โครงสร้างไวยากรณ์ไม่พึ่งบริบทประยุกต์เทคนิคทางสถิติ(PCFG) ขั้นตอนการแปลประโยคไทยล้านนาในขั้นตอนนี้จะเป็นการนำคำจากขั้นตอนก่อนการประมวลผลไปค้นหาในพจนานุกรมหากไม่พบคำแปลจะทำการแปลงรูปคำไทยล้านนาตามกฎการเขียนคำไทยล้านนา ในขั้นตอนสุดท้ายการสร้างประโยคไทยล้านนา เป็นการปรับโครงสร้างประโยคไทยให้เป็นโครงสร้างประโยคไทยล้านนา โดยผ่านไวยากรณ์ปริวรรตเพิ่มพูนและลดความกำกวมของโครงสร้างด้วยวิเทอร์บีอัลกอริทึม เมื่อผ่านกระบวนการทั้งหมดแล้วจะได้ประโยคไทยล้านนาที่เรียงลำดับตามความน่าจะเป็นของการเกิดประโยค สิ่งที่ผู้วิจัยได้จัดสร้างได้แก่ (1) พจนานุกรมอิเล็กทรอนิกส์ไทย-ไทยล้านนา (2) เครื่องจักรวิเคราะห์ไวยากรณ์ไม่พึ่งบริบท บริบทประยุกต์เทคนิคทางสถิติ (3) กฎการแปลงรูปภาษาไทยเป็นภาษาไทยล้านนา (4) กฎไวยากรณ์ปริวรรตเพิ่มพูนเพื่อจัดโครงสร้างประโยคไทยล้านนา สุดท้ายเรียงลำดับคำแปลที่น่าจะเป็นตามลำดับเมื่อคำนวณจากวิเทอร์บีอัลกอริทึม การทดสอบประสิทธิภาพในการแปลโดยใช้นิทานพื้นบ้านและประโยคทดสอบจำนวน 206 ประโยค นำผลลัพท์มาเปรียบเทียบกับประโยคไทยล้านนาจากผู้เชี่ยวชาญภาษาไทยล้านนา ปรากฏว่ามีประโยคที่แปลถูกต้อง 162 ประโยคจาก 206 ประโยคคิดเป็นอัตราส่วนร้อยละ 78.64

Thesis	Thai to Thai-lanna Machine Translation
Student	Mr.Prathan Comejina
Student ID.	51066409
Degree	Master of Science
Program	Information Technology
Major	Information Science
Year	2013
Thesis Advisor	Assoc.Prof. Dr. Pornrudee Netisopakul

ABSTRACT

This thesis proposes to translate Thai language to Thai-lanna language in a sentence level. There are three main natural language processes: preprocessing, mapping Thai to Thai-lanna, and Thai-lanna syntactic analysis. Preprocessing steps consist of two sub-steps, Thai word segmentation using a KUCUT tool and Thai sentence analysis using Probabilistic Context Free Gramma (PCFG). Mapping steps translate Thai words to Thai-lanna words using Thai-Thai-lanna dictionary. If a word is not found in the dictionary, we use Thai-lanna written rules to directly translate the Thai word. The final process constructs a Thai-lanna sentence using Thai-Thai lanna transformation rule. If ambiguity occurs – there can be more than one sentence structure- we sort the sentences based on their probabilities given by a viterbi algorithm. Our contributions are construction of (1) Thai-Thai lanna electronic dictionary in a trie structure, (2) a Thai PCFG grammar for parsing Thai sentences, (3) rules for Thai-Thai lanna word translation, (4) transformation rules for Thai-Thai lanna sentence use during Thai-lanna sentence generation, and finally, implementation of viterbi algorithm to sort a set of possible Thai-lanna sentences. We use our system to translate three folk stories and a set of general sentences, totally 206 sentences. The first sentence in the sorted translated list is judged by a Thai lanna expert to be translated correctly 162 sentences out of 206 sentences, or about 78.64%.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดีด้วยความกรุณาจาก รศ.ดร.พรฤดี เนติโสภากุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ได้ให้คำแนะนำ แนวคิด ตลอดจนแก้ไขข้อบกพร่องต่างๆ เป็นอย่างดี จนวิทยานิพนธ์ฉบับนี้เสร็จสมบูรณ์ ซึ่งข้าพเจ้าขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ บิดา มารดาผู้ซึ่งให้ความรักความเมตตาความห่วงใยและเป็นกำลังใจให้กับผู้จัดทำวิทยานิพนธ์ฉบับนี้จนสำเร็จ

ขอขอบพระคุณ คณาจารย์คณะเทคโนโลยีสารสนเทศ สถาบันพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาการตลอดระยะเวลาที่ได้ศึกษาในสถาบันแห่งนี้ รวมถึงคณะกรรมการที่ดำเนินการสอบที่ได้ให้คำแนะนำและคำชี้แนะในการปรับปรุงวิทยานิพนธ์ฉบับนี้ให้มีความสมบูรณ์

ขอขอบคุณพี่ๆ น้องๆ สมาชิกหน่วยปฏิบัติการการจัดการองค์ความรู้และวิศวกรรมความรู้ (KMAKE Lab) ทุกคนที่ได้ให้กำลังใจและช่วยเหลือในทุกด้านในการทำวิทยานิพนธ์ ขอบคุณ คุณกฤตาพร พัชระสุภา ที่ได้ให้ความช่วยเหลือให้คำแนะนำองค์ความรู้ในเรื่องการประมวลผลภาษาธรรมชาติเป็นอย่างดีในการทำวิทยานิพนธ์ฉบับนี้

ขอบคุณสาขาวิชาคอมพิวเตอร์ มหาวิทยาลัยราชภัฏเชียงใหม่ ที่ได้ให้เวลาในการทำวิทยานิพนธ์ รวมถึงเพื่อนร่วมงานทุกท่านที่ได้ให้กำลังใจเป็นอย่างดี

ความดีของการศึกษาค้นคว้าครั้งนี้ขอมอบเป็นเครื่องบูชาบิดา มารดาและบูรพาจารย์ทุกท่าน ผู้เขียนมีความซาบซึ้งในความกรุณาอันดียิ่งจากทุกท่านที่ได้กล่าวนามมาและขอขอบพระคุณมา ณ โอกาสนี้

ประธาน คำจินะ

สารบัญ

สารบัญ(ต่อ)

หน้า

สารบัญตาราง

สารบัญรูป

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ภาษาเป็นวิธีการในการสื่อสารของมนุษย์อีกวิธีการหนึ่ง ซึ่งสามารถสื่อสารกันโดยการเรียนรู้ความหมายและความต้องการของผู้สื่อสารจาก เสียงและสัญลักษณ์ที่แสดงหรือตัวอักษรนั่นเอง ในแต่ละชนชาติบนโลกนี้ก็ล้วนแต่มีวิธีการสื่อสารกันซึ่งมีความแตกต่างกันตามสังคม วัฒนธรรมรวมถึงภูมิประเทศ ประเทศไทยก็เป็นประเทศหนึ่งที่มีความหลากหลายทางวัฒนธรรม แต่ละภูมิภาคก็จะมีภาษาถิ่นที่มีเอกลักษณ์ นอกเหนือจากภาษาไทยกลาง ที่เป็นภาษาราชการของประเทศไทย นอกจากนี้ภาษาถิ่นแต่ละภาคก็มีความแตกต่างทางด้านโทนเสียงและสัญลักษณ์ ภาษาถิ่นเหล่านั้นเป็นภาษาที่แสดงออกถึงภูมิปัญญาของบรรพบุรุษ ที่จะใช้ตัวอักษรเหล่านั้น จารึกหรือบันทึก ข้อมูลที่สำคัญๆ ของอดีตจนถึงปัจจุบันเอาไว้

ภาษาไทยล้านนาเป็นภาษาถิ่นที่ใช้กันอย่างแพร่หลายในภาคเหนือตอนบนของประเทศไทย ภาษาไทยล้านนาเป็นภาษาที่มีลักษณะพิเศษเฉพาะตัว มีต้นกำเนิดมาจากภาษาขอม ภาษาไทยล้านนานั้นจะมีทั้งภาษาพูดและภาษาเขียน คนเมืองหรือคนล้านนาจะพูดและเขียนภาษาคำเมือง , ตัวธรรมหรือภาษาไทยล้านนาแล้วแต่จะเรียก ซึ่งเป็นภาษาหลักในการสื่อสาร ในการบันทึกเรื่องราวเหตุการณ์ต่างๆ ตำรายาสมุนไพร หรือแม้กระทั่งคัมภีร์ใบลาน กัณฑ์เทศ จารึก ก็ใช้อักษรไทยล้านนาเป็นอักษรในการสื่อถึงเหตุการณ์เหล่านั้น ในปัจจุบันเมื่อมีความก้าวหน้าทางสังคมและเทคโนโลยี รวมไปทั้งมีหลักสูตรทางการเรียนการสอนที่เป็นมาตรฐานของไทย ทำให้คนเมืองหรือคนล้านนารุ่นหลังได้ปรับเปลี่ยนวิถีชีวิต ภาษาคำเมืองหรือภาษาไทยล้านนา ก็เริ่มถูกกลืนหายไปเหลือเพียงบางพื้นที่ยังใช้ภาษาพูด ในการติดต่อสื่อสารกัน ส่วนการเรียนรู้ตัวอักษร หรือภาษาเขียนนั้น จะมีแต่ในวงแคบเช่น วัดหรือในบางสถานศึกษาเท่านั้นที่มีการจัดการเรียนการสอนภาษาไทยล้านนา จึงควรค่าสืบสานไม่ให้สูญหาย

การแปลภาษาด้วยเครื่องคอมพิวเตอร์เป็นเทคโนโลยีหนึ่งที่ใช้โปรแกรมทางคอมพิวเตอร์แปลจากภาษาหนึ่งไปเป็นภาษาเป้าหมาย ซึ่งในการแปลภาษานั้นมีวิธีการแปลได้หลายแบบ เช่น การแปลแบบคำต่อคำ การแปลแบบวิเคราะห์โครงสร้างไวยากรณ์ และการแปลโดยใช้ภาษากลาง การแปลแต่ละแบบก็สามารถประยุกต์เข้ากับแต่ละข้อจำกัดของภาษา

จากเหตุผลข้างต้นจึงทำให้เกิดแนวคิดงานวิจัย การแปลภาษาไทยเป็นภาษาไทยล้านนา โดยวิเคราะห์โครงสร้างของไวยากรณ์ของภาษาไทยล้านนา และค้นหาคำศัพท์ในพจนานุกรมไทย-ล้านนา ที่สร้างขึ้นแล้วก็จะลดความกำกวมของประโยคด้วยวิธีการของแบบจำลองฮิดเดนมาร์คอฟใช้วิเทอร์บีอัลกอริทึม และใช้ค่าความน่าจะเป็นจากคลังประโยคไทยล้านนา

1.2 วัตถุประสงค์ของการศึกษา

1.2.1 เพื่ออนุรักษ์ภาษาไทยล้านนาที่ปัจจุบันมีผู้เชี่ยวชาญในภาษาไทยล้านนาลดน้อยลงไม่ให้สูญหาย

1.2.2 เพื่อศึกษาและพัฒนาแนวทางในการเลือกวิธีการแปลภาษา เพื่อใช้ในการแปลภาษาที่มีความใกล้เคียงกัน

1.2.3 เพื่อพัฒนาต้นแบบโปรแกรมประยุกต์ในการแปลภาษาไทยเป็นภาษาไทยล้านนา

1.3 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

1.3.1 วิธีการปริวรรตภาษาไทยล้านนาเป็นการเขียนและวางตำแหน่งตัวอักษรไทยล้านนา ตามรูปแบบ โดยจะมีรูปแบบตามวิธีการเขียนภาษาไทยล้านนาของ จ.เชียงใหม่ ซึ่งช่วยในการแปลงรูปของคำภาษาไทยที่ค้นหาในพจนานุกรมแล้วไม่เจอคำดังกล่าว ซึ่งจะทำให้ได้รูปของคำภาษาไทยล้านนา

1.3.2 การวิเคราะห์ชนิดของคำภาษาไทย เพื่อนำไปวิเคราะห์โครงสร้างไวยากรณ์และเปรียบเทียบโครงสร้างกับไวยากรณ์ภาษาไทยล้านนา และใช้ในการหาคำศัพท์ไทยล้านนาที่ตรงกับประเภทของคำไทยที่ต้องการ

1.3.3 การวิเคราะห์โครงสร้างไวยากรณ์ภาษาไทยด้วย PCFG

1.3.4 แบบจำลองฮิดเดนมาร์คอฟโมเดลและวิเทอร์บีอัลกอริทึม จะใช้ในการคำนวณลำดับของประโยคที่มีความน่าจะเป็นสูงสุด

1.4 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1.4.1 ได้ระบบการแปลภาษาไทยเป็นภาษาไทยล้านนาเพื่อสนับสนุนการเรียนรู้ภาษาไทยล้านนาสำหรับผู้ที่สนใจ

1.4.2 สามารถเผยแพร่ภูมิปัญญาท้องถิ่นและอนุรักษ์วัฒนธรรมไทย ออกไปสู่วงกว้างซึ่งจะช่วยให้องค์ความรู้ทางด้านภาษาไทยล้านนาถูกสืบสานต่อไป

1.4.3 ได้แนวทางในการแปลภาษาไทยเป็นภาษาอื่นที่มีลักษณะ ข้อจำกัดของภาษาที่มีความใกล้เคียงกันกับภาษาไทยล้านนา

1.5 ขอบเขตการวิจัย

1.5.1 พจนานุกรมที่ใช้ในงานวิจัยคือ พจนานุกรมล้านนา-ไทย ฉบับเฉลิมพระเกียรติพระบาทสมเด็จพระเจ้าอยู่หัว ในวโรกาสเจริญพระชนมายุ 80 พรรษา ของสถาบันภาษาและศิลปวัฒนธรรม มหาวิทยาลัยราชภัฏเชียงใหม่

1.5.2 การแปลภาษาไทยล้านนาจะอยู่ในระดับประโยค

1.5.3 ในขั้นตอนการลดความกำกวมโดยใช้วิเทอร์บีอัลกอริทึม จะใช้คลังข้อมูลภาษาที่เป็นภาษาไทยล้านนา จากประโยคภาษาไทยล้านนาที่รวบรวมได้

1.6 ขั้นตอนของการศึกษา

ขั้นตอนในการศึกษาวิทยานิพนธ์มีขั้นตอนดังต่อไปนี้

1.6.1 ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้อง

1.6.2 ออกแบบขั้นตอนการแปลภาษาไทยเป็นภาษาไทยล้านนา

1.6.3 สร้างพจนานุกรมไทย-ล้านนา จากพจนานุกรมล้านนา-ไทย ฉบับเฉลิมพระเกียรติพระบาท สมเด็จพระเจ้าอยู่หัว ในวโรกาสเจริญพระชนมายุ 80 พรรษา ของสถาบันภาษาและศิลปวัฒนธรรม มหาวิทยาลัยราชภัฏเชียงใหม่

1.6.4 เตรียมข้อมูลคลังภาษาไทยล้านนา เพื่อใช้ในการเรียนรู้ในการลดความกำกวมโดยใช้วิเทอร์บีอัลกอริทึม

1.6.5 สร้างเครื่องมือต้นแบบ การแปลภาษาไทยเป็นภาษาไทยล้านนา

1.6.6 ทดลองการแปล เก็บผลลัพท์ของการแปล และให้ผู้เชี่ยวชาญช่วยตรวจสอบผลลัพธ์ที่ถูกต้องของคำแปลไทยล้านนาที่ได้จากการแปล

1.6.7 วิเคราะห์ผลผลลัพธ์และสรุปผลการทดลอง

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีพื้นฐานที่ใช้ในการวิจัย และงานวิจัยที่เกี่ยวข้องซึ่งจะเป็นพื้นฐานของงานวิจัยการแปลภาษาไทยเป็นภาษาไทยล้านนา ทฤษฎีที่เกี่ยวข้องกับงานวิจัยได้แก่ ภาษาไทยล้านนา งานวิจัยที่เกี่ยวข้อง

2.1 พื้นฐานของภาษาไทยล้านนา [1]

ในการศึกษาภาษาไทยล้านนานั้น จะต้องทราบถึงรูปของตัวอักษร ก่อนเพื่อนที่จะเป็นพื้นฐานในการเขียนและการอ่าน รูปของภาษาไทยล้านนาสามารถแบ่งออกได้ดังนี้

2.1.1 รูปพยัญชนะ อักษรไทยล้านนา หรือตั๋วเมือง จัดตามกลุ่มภาษาบาลีแบ่งออกเป็น 5วรรค วรรคละ 5 ตัวอักษร ซึ่งเรียกว่า พยัญชนะวรรค หรือ พยัญชนะในวรรค และมีอีก 8 ตัวอักษรที่ไม่จัดอยู่ในวรรค เรียกว่า พยัญชนะอวรรค หรือ พยัญชนะนอกวรรค

พยัญชนะในวรรค มีทั้งหมด 5 วรรค 25 คือ

กะ วรรค	กะ	ขะ	ก๊ะ	ฆะ	งะ
	ᨠ	ᨡ	ᨣ	ᨥ	ᨦ
จะ วรรค	จะ	ฉะ	จ๊ะ	ฌะ	ญะ
	ᨧ	ᨨ	ᨩ	ᨫ	ᨬ
ระฏะ วรรค	ระฏะ	ระฐะ	ฑะ	ระฒะ	ระณะ
	ᨭ	ᨮ	ᨯ	ᨰ	ᨱ
ตะ วรรค	ตะ	ถะ	ต๊ะ	ธะ	นะ
	ᨲ	ᨳ	ᨴ	ᨵ	ᨶ
ปะ วรรค	ปะ	ผะ	ป๊ะ	ภะ	มะ
	ᨸ	ᨹ	ᨻ	ᨽ	ᨾ

พยัญชนะนอกวรรค มีทั้งหมด 8 ตัวอักษรได้แก่

ยะ	ระ	ละ	วะ	สะ	หะ	ฬะ	อัง
ᨿ	ᩁ	ᩃ	ᩅ	ᩈ	ᩉ	ᩊ	ᩋ

พยัญชนะเพิ่ม พยัญชนะยังไม่ครบตามสำเนียงภาษาพูด คนโบราณเลยเพิ่มพยัญชนะอีก 10 ตัว คือ

คะ	ซะ	บะ	ยะ (อยะ,หยะ)	ฝะ
ᨤ	ᨪ	ᨷ	ᨿ	ᨺ
ฟะ	ศะ	ฮะ	สะสองห้อง	นา
ᨼ	ᩆ	ᩌ	ᩔ	ᩉ

พยัญชนะพิเศษ เป็นพยัญชนะที่มีใช้บ้างแต่ก็เป็นบางคำเท่านั้นไม่ค่อยได้ใช้บ่อยนัก

ᨻ ตั๋วป๊ะหลวง มีค่าเท่ากับตัว พ และ พพ เช่น สพฺพํ(ᩈᨻ᩠ᨻᩴ)

ᩂ ตั๋ว รือ, ฤา เช่น ฤกษ์(ᩂᨠ᩠ᩇ)

ᩕ ระโฮง เท่ากับตัว ร ควบกล้ำ เช่น พระ(ᨻᩕ) พรหมา(ᨻᩕᩉ᩠ᨾᩣ)

ᩡ ไม้ กะ (สระอะ) บางครั้งเท่ากับตัว ก สะกด เรียกว่า กะปู่ยาด เช่น (ᨲᩪᩡ)

2.1.2 รูปสระ สระในภาษาล้านนานั้นแบ่งออกเป็นหลายประเภท คือ สระตามภาษาบาลี สระเดี่ยว และสระผสม ซึ่งต่างก็มีลักษณะการใช้งานและหน้าที่แตกต่างกันไป

สระที่มาจากภาษาบาลี

อะ	อา	อิ	อี	อุ	อู	เอ้	โอ้
ᩋ	ᩋᩣ	ᩍ	ᩎ	ᩏ	ᩐ	ᩑ	ᩒ

สระเดี่ยว เป็นสระแท้ (ไม่สามารถออกเสียงได้ด้วยตัวเอง)เรียกว่า “ไม้” ใช้ผสมกับพยัญชนะมี 14 ตัว

-ะ	-ᩡ	ไม้กะ	-า	-ᩣ	ไม้ก๋า
-ิ	-ᩥ	ไม้กิ	-ี	-ᩦ	ไม้กี๋
-ึ	-ᩧ	ไม้กึ	-ื	-ᩨ	ไม้กือ
-ุ	-ᩩ	ไม้กุ	-ู	-ᩪ	ไม้กู๋
เ-	ᩮ-	ไม้เก๋	แ-	ᩯ-	ไม้แก๋
ไ-	ᩱ-/ᩲ-	ไม้ไก๋	โ-	ᩰ-	ไม้โก๋
-ำ	-ᩴᩣ	ไม้ก๋ำ	นิคหิต		ไม้ก๋อ, ไม้กั๋งมน (บาลี)

สระผสม เป็นการนำเอาสระแท้ หรือสระเดี่ยวมาผสมรวมกัน เรียกว่าการผสมไม้ ตัวอย่างเช่น

ᨠᩮᩡ	ไม้เกะ	ᨠᩯᩡ	ไม้แกะ
ᨠᩰᩡ	ไม้โกะ	ᨠᩮᩣ	ไม้โก๋(บาลี)
ᨠᩮᩢᩣ	ไม้เก๋า	ᨠᩬᩴᩡ	ไม้เกาะ
ᨠᩬᩴ	ไม้ก๋อ	ᨠ᩠ᩅᩡ	ไม้กั๊ะ
ᨠ᩠ᩅ	ไม้กั๋ว	ᨠᩮ᩠ᨿᩡ	ไม้เกียะ
ᨠᩮ᩠ᨿ	ไม้เกี๋ย	ᨠᩮᩨᩬᩡ	ไม้เกือะ
ᨠᩮᩨᩬ	ไม้เกื๋อ	ᨠᩮᩥᩬᩡ	ไม้เกอะ
ᨠᩮᩥᩬ	ไม้เก๋อ	ᨠᩱ᩠ᨿ	ไม้ไก๋ย
ᨠᩱ᩠ᨿ᩠ᨿ	ไม้ไก๋ยย	ᨠᩮᨿ᩠ᨿ	ไม้ไก๋ยย (บาลี)

2.1.3 รูปวรรณยุกต์/สัญญลักษณ์พิเศษ อดีตไม่มีวรรณยุกต์เหล่านี้ปรากฏซึ่งในใบลานเก่า จะพบบ่อยเมื่อสุดปลายใบลาน

᩵	ไม้เอก	ไม้หยัก, ไม้เหยาะ
᩶	ไม้โท	ไม้ขอช้าง
᩷	ไม้หันอากาศ	ไม้ซัด
᩼	การันต์	ระห้าม
᩸	ไม้ซ้ำคำ	ไม้สองน้อย
᩹	ไม้ไต่คู้	ไม้ไต่คู้
ᩴ	ไม้ก๋ง,ตั๋วข่ม, ไม้โก๋(ลดรูป)	
᩺		ไม้กั๋งไหล
᩻		ไม้เก๋าจู้จี้
᩼		ไม้เก๋าห่อหนึ้ง

2.1.4 พยัญชนะตัวสะกด

หลักในการใช้พยัญชนะสะกดของภาษาไทยล้านนา มีวิธีการใช้ตามลักษณะกฎเกณฑ์ดังนี้

- ตัวสะกดอยู่ด้านล่าง เช่น ᨠ᩠ᨶ ᨸ᩠ᨶ ᨷ᩠ᨶ
- เมื่อไม้หรือสระหรือพยัญชนะควบกล้ำอยู่ข้างข้างล่างตัวพยัญชนะ มักนิยม เขียนตัวสะกดอยู่ถัดไปแนวเดียวกันเช่น ᩃᩪᨠ ᨠᩩᨾ ᨲᩪᨠ

- ถ้าพยัญชนะตัวสะกดเป็น พยัญชนะเหล่านี้ให้ใช้หางของตัวสะกดแทนและอยู่ข้างล่าง หรือห้อยด้านข้างพยัญชนะเสมอตามลักษณะของแต่ละตัว ยกเว้นมีไม้หรือพยัญชนะควบกล้ำอยู่ใต้ตัวอักษร

ตัว น สะกด ᨶ ให้ใช้ ᩠ᨶ สะกด

ตัว บ สะกด ᨷ ให้ใช้ ᩠ᨷ สะกด

ตัว ป สะกด ᨸ ให้ใช้ ᩠ᨸ สะกด

ตัว พ สะกด ᨻ ให้ใช้ ᩠ᨻ สะกด

ตัว ผ สะกด ᨹ ให้ใช้ ᩠ᨹ สะกด

ตัว ม สะกด ᨾ ให้ใช้ ᩠ᨾ สะกด

ตัว ย สะกด ᨿ ให้ใช้ ᩠ᨿ สะกด

2.1.5 หลักการเขียนภาษาไทยล้านนา [2]

หลักการเขียนพยัญชนะไทยล้านนาจะมีความแตกต่างจากหลักภาษาไทย เพราะพยัญชนะไทยล้านนาเมื่อเป็นตัวสะกด บางทีจะห้อยอยู่ด้านล่าง ด้านข้างหรือเปลี่ยนรูป ซึ่งมีวิธีใช้ตามลักษณะกฎเกณฑ์ดังนี้

- ตัวสะกดโดยปกติจะอยู่ด้านล่าง

- เมื่อมีไม้หรือสระ หรือพญัชนะควบกล้ำอยู่ด้านล่างตัวพยัญชนะ มักนิยมเขียนตัวสะกดอยู่ถัดไปในแนวเดียวกัน

- ถ้าพยัญชนะตัวสะกดเป็น ᨶ ᨷ ᨸ ᨻ ᨹ ᨾ ᨿ ให้ใช้หางของตัวเองสะกดแทนและอยู่ข้างล่างหรือห้อยด้านข้างพยัญชนะเสมอ(ตามลักษณะของแต่ละตัว) ยกเว้นมีไม้หรือพยัญชนะควบกล้ำอยู่ใต้ตัวอักษร

วิธีใช้สระและไม้ต่าง ๆ

- ᨲᩡ ไม้กะ (สระอะ) คำที่ออกเสียงสระอะ จะไม่เติมไม้กะลงไป เช่น จะไป ᨧᨷᩱ คะเน ᨤᨶᩮ ตะวัน ᨲᩅᩢ᩠ᨶ เมื่ออยู่ท้ายพยัญชนะ จะเท่ากับตัวไม้ไต่คู้ คือบังคับให้ออกเสียงสั้น เร็ว เช่น เห็ด ᩉᩮ᩠ᨯᩡ

- ᨲᩩ ไม้กุ (สระอุ) ไม้กุ โดยทั่วไปก็จะใช้เหมือนปกติ แต่จะมีข้อน่าสังเกตที่เวลามีพยัญชนะอยู่ด้านล่าง จะเขียนอยู่ด้านล่างหรือด้านข้างก็ได้

- ᩮ- ไม้โก⁺ (สระโอ) ใช้เหมือนสระโอ แต่ถ้าเป็นคำที่มาจากภาษาบาลี หรือ คำบาลีจะใช้ ᩮ-ᩣ แทนไม้โก๋ เช่น คำว่า พุทฺโธ ธมฺโม สงฺโฆ โหติ ᨻᩩᨴ᩠ᨵᩮᩣ ᨵᨾ᩠ᨾᩮᩣ ᩈᩘ᩠ᨥᩮᩣ ᩉᩮᩣᨲᩥ

- ᩮ-ᩭ ไม้เกี๋ย การใช้ไม้เกี๋ย มีที่ต้องสังเกตคือ หากไม่มีตัวสะกด เช่น เมีย เสีย ก็จะเขียนว่า ᩮᨾᩭ ᩮᩈᩭ เมื่อมีตัวสะกด ก็จะละ ᩮ คงไว้แต่ ᩭ ต่อด้วยตัวสะกด เช่น เสียง เวียง ᩈ᩠ᨿᨦ᩺ ᩅ᩠ᨿᨦ᩺

- ᩱ-ᩭ ไม้ไก๋ย ไม้ไก๋ย นี้ มีค่าเท่ากับตัว ไ- ย และ -ั ย เช่น ชัย ᩱᨩ᩠ᨿ วิทยาลัย ᩅᩥᨴ᩠ᨿᩣᩱᩃ᩠ᨿ ไทย ᩱᨴ᩠ᨿ

- ᩉ᩠ᨠ᩺ ซึ่งจะเป็นการใช้เมื่อมีพยัญชนะสองตัวซ้อนกันเสมอ ใช้เขียนบนพยัญชนะเพื่อให้รู้ว่าเป็นตัวสะกด เช่น ลูก ᩃᩪᨠ᩺ กอด ᨠᩬᨯ᩺ เสียม ᩈ᩠ᨿᨾ᩺

- -᩻ ไม้สองน้อย ใช้เหมือนไม้ซ้ำคำ แต่เขียนด้านบนพยัญชนะ เช่น ไป ๆ มา ๆ ᩱᨸ᩻ ᨾᩣ᩻ ใช้เขียนเพื่อให้รู้ว่าต้องอ่านแยก เช่น สนาม ᩈ᩻ᨶᩣᨾ สมัย ᩱᩈ᩻ᨾ

- -ᩴ ไม้กั๋ง, ไม้งาม, ตั๋วข่ม, ไม้โก๋ลดรูป ใช้เขียนบนพยัญชนะเพื่อบอกให้รู้ว่าเป็นแม่ กก กม กน กบ เช่น ᨠᩴ ᨠᩴ ᨠᩴ ᨠᩴ แต่ถ้าตัวสะกดเป็นตัว ว ก็จะออกเสียงเป็น อัว แทน เช่น ᨹᩴ ᨲᩴ

- ᩮ-ᩫᩣ ไม้เก๋าจู้จี้ เป็นรูปพิเศษ เท่ากับ เ-า เช่น เรา ᩮᩁᩫᩣ

- -ᩮᩢᩣ ไม้เก๋าห่อหนึ้ง เป็นรูปพิเศษ เท่ากับ เ-า เช่น เจ้า ᨧᩮᩢ᩶ᩣ

2.2 การแปลภาษา

การแปลภาษาเป็นศาสตร์หนึ่งที่ทำให้ซอร์ฟแวร์คอมพิวเตอร์ทำการแปลข้อความหรือคำพูดในภาษาธรรมชาติ จากภาษาหนึ่งที่เรียกว่าภาษาต้นทางไปยังอีกภาษาหนึ่งที่เรียกว่าภาษาเป้าหมาย โดยที่สามารถจำแนกการแปลภาษาจากองค์ประกอบในประโยค ได้ดังต่อไปนี้

2.2.1 การแปลภาษาระดับคำ

"คำ"[3] ประกอบขึ้นด้วยส่วนย่อยที่เรียกว่าหน่วยคำ(morpheme) คำหนึ่งอาจจะประกอบขึ้นด้วยหน่วยคำเพียง 1 หน่วยหรือมากกว่าก็ได้ ซึ่งในภาษาไทยภาษาไทยเป็นคำโดด การวิเคราะห์หน่วยคำในลักษณะแยกพยางค์จึงไม่เหมาะสมกับภาษาไทย ต่างจากภาษาอังกฤษที่มีการเติมพยัญชนะลงไปท้ายคำจะทำให้หน่วยคำดังกล่าวทำหน้าที่แตกต่างกันไป เช่น เมื่อเติม s ก็จะทำให้คำนั้นกลายเป็นพหูพจน์

การแปลภาษาจากภาษาต้นทางไปยังภาษาเป้าหมายวิธีการที่ง่ายที่สุดในระดับคำคือ ใช้วิธีการแปลแบบตรงไปตรงมา โดยเทียบความหมายแบบคำต่อคำ อย่างไรก็ตามในการแปลภาษาในระดับคำแบบตรงไปตรงมานั้น จะเหมาะสมกับบางภาษาที่มีโครงสร้างของไวยากรณ์ของประโยคที่มีความใกล้เคียงกัน ก็จะสามารถใช้วิธีการดังกล่าวได้ แต่ก็จะเกิดปัญหาในกรณีที่โครงสร้างของไวยากรณ์ของภาษาต้นทางและภาษาเป้าหมายมีความแตกต่างกัน ซึ่งโดยส่วนใหญ่แล้วภาษาในแต่ละชาติแต่ละท้องถิ่น จะมีความแตกต่างกันพอสมควร

ในการศึกษางานวิจัยการแปลภาษาล้านนาเป็นภาษาไทย[4] พบว่างานวิจัยนี้มีเป้าหมายในการแปลภาษาล้านนาให้เป็นภาษาไทยซึ่งเป็นคู่ภาษาเดียวกันกับงานวิจัยการแปลภาษาไทยเป็น

ภาษาไทยล้านนา งานวิจัยดังกล่าวได้นิยามว่าในการแปลภาษาไทยล้านนาเป็นภาษาไทยนั้น โครงสร้างไวยากรณ์ของประโยคของภาษาไทยล้านนาที่เป็นภาษาต้นทาง เป็นโครงสร้างเดียวกันกับสร้างไวยากรณ์ของประโยคของภาษาไทยที่เป็นภาษาเป้าหมายในการแปล ซึ่งจากเหตุผลขั้นต้นในงานวิจัยดังกล่าวจึงได้เลือกใช้วิธีการแปลในระดับคำด้วยวิธีการแบบตรงไปตรงมา โดยการใช้วิธีการค้นหาคำที่ได้จากประโยคต้นทางที่ผ่านการตัดคำแล้ว นำคำที่ต้องการแปลมาค้นหาคำแปลจากพจนานุกรมภาษาไทยล้านนา-ภาษาไทยที่ได้เตรียมไว้ ส่วนคำไหนที่ไม่พบในพจนานุกรมก็จะใช้ข่ายงานเพิ่มขยายในการสร้างหน่วยคำ งานวิจัยการแปลภาษาไทยล้านนาเป็นภาษาไทยเป็นงานที่มีความใกล้เคียงกับงานวิจัยการแปลภาษาไทยเป็นภาษาไทยล้านนาเพราะใช้คู่ภาษาเดียวกันคือภาษาไทยล้านนาและภาษาไทย แต่จะสลับภาษาต้นทางและภาษาเป้าหมายกัน นอกจากนั้นยังใช้วิธีการแปลโดยอ้างอิงพจนานุกรมคู่ภาษาเดียวกันอีกด้วย

สถาปัตยกรรมในการแปลภาษาไทยล้านนาเป็นภาษาไทย ในงานวิจัยดังกล่าวมีขั้นตอนที่เกี่ยวข้องกับโครงสร้างระดับคำ ดังต่อไปนี้ พจนานุกรมล้านนา-ไทย ที่ใช้ในงานวิจัยการแปลภาษาล้านนาเป็นภาษาไทย เป็นพจนานุกรมภาษาล้านนาเป็นภาษาไทย มีจำนวนคำทั้งหมด 7,497 คำ มีองค์ประกอบ 3 ส่วน คือ

- คำล้านนา
- เครื่องหมายกำกับ
- ส่วนวิเคราะห์ความหมาย ประกอบด้วย ประเภทของคำ คำที่ถอดออกมาเป็นภาษาไทย ส่วนกำกับหมวดคำ ลักษณะประจำคำ ความหมายภาษาอังกฤษ และสัทอักษร

การจัดเก็บ จัดเก็บเป็นโครงสร้างแบบทรัย มีลักษณะคล้ายโครงสร้างข้อมูลแบบต้นไม้ โดยมีเส้นเชื่อมของโหนด โดยที่แต่ละโหนดจะแสดงตำแหน่งของตัวอักษรถัดไป การค้นคืนของโครงสร้างแบบทรัยจะทำได้รวดเร็วแต่ต้องใช้หน่วยความจำในการจัดเก็บมาก จุดเด่นคือ สะดวกในการควบคุมข้อมูลที่มีความยาวแตกต่างกัน มีความรวดเร็วในการเข้าถึงข้อมูลเพราะเข้าถึงข้อมูลโหนดหนึ่งไปยังโหนดถัดไป

วิธีการในการแปลเมื่อรับข้อมูลต้นทางที่เป็นประโยคภาษาไทยมาล้านนา ก็จะมีขั้นตอนประกอบไปด้วย การตัดคำ การเลือกคำแปลในพจนานุกรมและการปริวรรตคำไทยจากคำล้านนาที่ไม่พบในพจนานุกรม

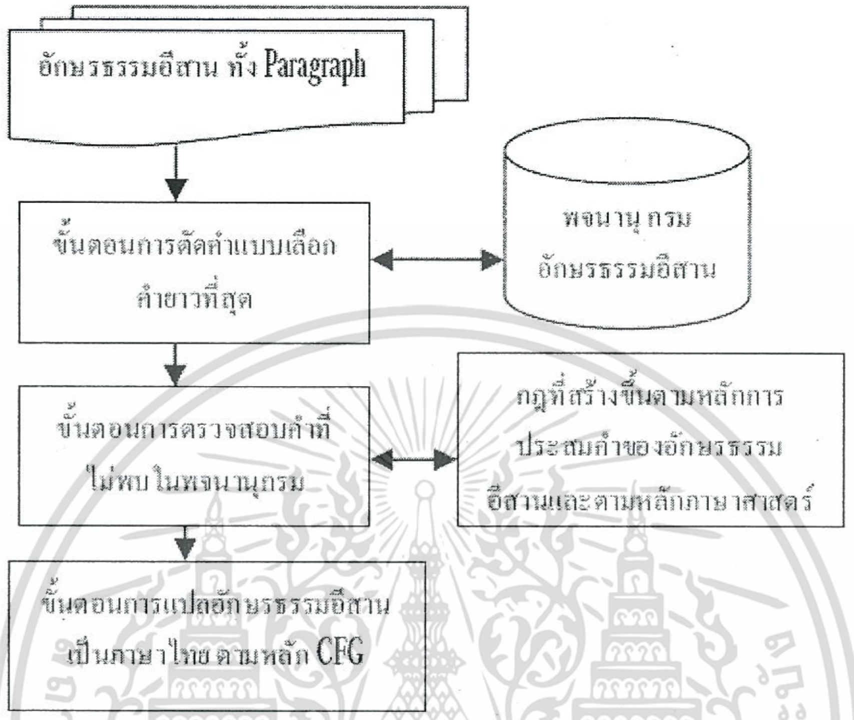
- ในขั้นตอนของการตัดคำจะใช้วิธีการตัดคำล้านนาที่นำเข้ามาแบบเลือกคำยาวที่สุด (Longest matching) โดยจะนำประโยคที่ต้องการตัดคำ ตัดตัวอักษรทางขวาออกทีละ 1 ตัวอักษร จากนั้นจะนำอักษรที่เหลือทางด้านซ้ายมือไปค้นหาในพจนานุกรม เพื่อหารูปแบบของคำที่ตรงกับคำที่เหลือทางซ้ายมือ เมื่อค้นหาพบก็จะสามารถตัดพยางค์และได้คำที่ตรงกับพจนานุกรม ผลลัพท์จะได้คำที่ผ่านการตัดคำจากประโยคนำเข้า

- การปริวรรตคำที่ไม่พบในพจนานุกรม ในงานวิจัยดังกล่าวจะใช้ข่ายงานเพิ่มขยาย (Nondeterministic Finite Automata NFA) สร้างกฎการปริวรรตล้านนาเป็นอักษรไทยเพื่อทำการแปลงคำไทยล้านนาที่ตัดคำและนำไปค้นหาในพจนานุกรมแล้งปรากฏว่าไม่พบข้อมูลคำไทยล้านนาคำดังกล่าวในพจนานุกรม มาแปลงรูปจากโครงสร้างของคำภาษาไทยล้านนาให้เป็นโครงสร้างคำภาษาไทย

ภาพที่ 2.1 : สถาปัตยกรรมในการแปลภาษาไทยล้านนาเป็นภาษาไทย

นอกจากนั้นในการศึกษางานวิจัยการแปลภาษาอักษรธรรมอีสานให้เป็นภาษาไทย[5] งานวิจัยนี้เป็นงานวิจัยที่ใกล้เคียงกับงานวิจัยที่สนใจตรงที่เป็นงานวิจัยที่เป็นภาษาถิ่น นอกจากนั้นอักษรธรรมอีสานยังมีลักษณะและรูปของตัวอักขระที่ใกล้เคียงกับอักษรไทยล้านนาเนื่องจากในอดีต อาณาจักรล้านช้างที่เป็นอาณาจักรของชนชาติที่อยู่ติดกับลุ่มน้ำโขง กับอาณาจักรล้านนาที่เป็นอาณาจักรของหลายจังหวัดภาคเหนือตอนบนของประเทศไทย นั้นได้รับอิธิพลวัฒนธรรมจากอาณาจักรขอมซึ่งรวมไปถึงตัวอักษรที่ใช้จดบันทึกด้วย งานวิจัยดังกล่าวในระดับคำนั้นได้ใช้วิธีการคล้ายกับงานวิจัยในการแปลภาษาไทยล้านนาเป็นภาษาไทยโดย จะนำคำที่ผ่านการตัดคำมาค้นหาในพจนานุกรมที่ได้เตรียมไว้ หากไม่พบคำดังกล่าวก็จะทำการแปลงอักษธรรมอีสานแปลงรูปให้เป็นภาษาไทยด้วยกฎที่สร้างขึ้นตามหลักการประสมคำของอักษรธรรมอีสาน ตามหลักภาษาศาสตร์

ซึ่งมีความคล้ายคลึงกับงานวิจัยการแปลภาษาไทยล้านนาเป็นภาษาไทย ในงานวิจัยดังกล่าวมีขั้นตอนดังต่อไปนี้



ภาพที่ 2.2 : สถาปัตยกรรมของการแปลอักษรธรรมอีสานให้เป็นภาษาไทย

วิธีการในการแปลอักษรธรรมอีสานให้เป็นภาษาไทยเมื่อรับข้อมูลต้นทางที่เป็นอักษรธรรมอีสาน ก็จะมีขั้นตอนประกอบไปด้วย การตัดคำ การเลือกคำแปลในพจนานุกรมและการสร้างคำที่ไม่พบในพจนานุกรม

- ในขั้นตอนของการตัดคำจะใช้วิธีการตัดแบบเลือกคำยาวที่สุด(Longest matching) วิธีการตัดคำแบบเลือกคำยาวมีวิธีคล้ายกับงานวิจัยการแปลภาษาไทยล้านนาเป็นภาษาไทย ผลลัพท์ก็จะได้อักษรธรรมอีสานในแต่ละคำจากนั้นก็จะนำคำที่ได้ไปค้นหาคำแปลในพจนานุกรมอักษรธรรมอีสานผลลัพท์จะได้คำแปลที่เป็นภาษาไทย

- พจนานุกรมอักษรธรรมอีสานในการแปลอักษรธรรมอีสานให้เป็นภาษาไทย ก็จะทำการสร้างพจนานุกรมอักษรธรรมอีสาน มีการจัดเก็บเป็นโครงสร้างแบบทรัย เพื่อใช้ในการเลือกคำแปลอักษรธรรมอีสานให้เป็นภาษาไทย

- เมื่อนำคำที่ต้องการไปค้นในพจนานุกรมแล้ว หากไม่พบคำศัพท์ดังกล่าวในพจนานุกรมอักษรธรรมอีสาน ก็จะนำคำอักษรธรรมอีสานที่ไม่พบในพจนานุกรมนั้นเข้าสู่ขั้นตอน

การประสมคำจากหลักการประสมคำของอักษรธรรมอีสานให้เป็นภาษาไทยตามหลักภาษาศาสตร์ โดยใช้กฎที่สร้างขึ้นตามหลักการประสมคำอักษรธรรมอีสาน

- ผลลัพท์จากขั้นตอนการค้นหาคำศัพท์และการประสมคำจากหลักการประสมคำของอักษรธรรมอีสานให้เป็นภาษาไทย ก็จะประโยคภาษาไทยที่ยังไม่ใช่โครงสร้างไวยากรณ์ภาษาไทย ต้องผ่านขั้นตอนของการแปลโดยใช้หลักวิเคราะห์โครงสร้างไวยากรณ์ไม่พึ่งบริบท (Context-free grammar: CFG)

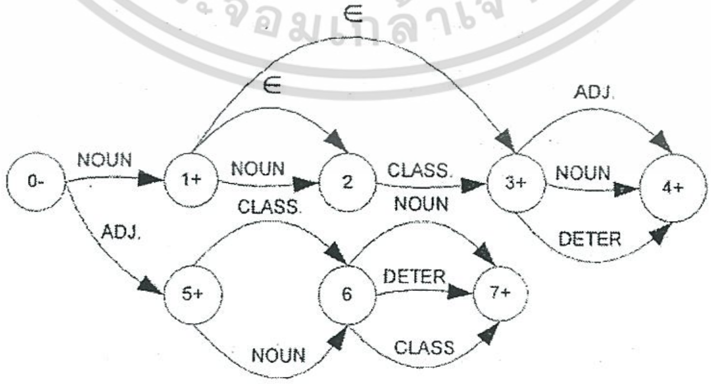
2.2.2 การแปลภาษาระดับวลี

วลี (Phrases) [3] คือกลุ่มคำที่ประกอบด้วยคำต่างๆ ที่นำมาเรียงกันอย่างมีความหมายและทำหน้าที่ใดหน้าที่หนึ่งในประโยค เช่นเป็นประธาน กริยา กรรม ส่วนเสริม ส่วนกริยาวิศษณ์ วลีไม่ได้ประกอบด้วยภาคประธานและภาคแสดง นักภาษาศาสตร์อย่าง Chomsky ได้กล่าวว่า คำเพียงหนึ่งคำก็เป็นวลี แต่ในที่นี้จะถือว่า วลีคือคำตั้งแต่ 2 คำขึ้นไปที่ทำหน้าที่ใดหน้าที่หนึ่งในประโยค จากงานวิจัยการแปลภาษาล้านนาเป็นภาษาไทย[4] ในขั้นตอนของการวิเคราะห์โครงสร้างวลีและโครงสร้างประโยค จะใช้ข่ายงานเพิ่มขยาย Augmented Transition Network (ATNs) ซึ่งเป็นกราฟชนิดหนึ่งประกอบด้วยสถานะซึ่งแทนด้วยรูปวงกลมและการส่งผ่านค่าซึ่งแทนด้วยเส้นโค้ง สถานะสุดท้ายจะกำกับด้วย “/1” ในแต่ละเส้นโค้งจะมีตัวเก็บค่าและหมายเลขกำกับเพื่อส่งผ่านคำในประโยคเข้าสู่ข่ายงาน ตัวเก็บค่าจะทำหน้าที่วิเคราะห์และบันทึกบทบาทของคำตามเงื่อนไขที่สร้างขึ้น

เมื่อนำประโยคมาผ่านโครงสร้างเพื่อหาความสัมพันธ์ในประโยคแล้วก็จะได้ความสัมพันธ์ของวลีในประโยคภาษาตั้งต้น โครงสร้างวลีของภาษาไทย จะประกอบด้วย 2 ส่วนคือส่วนมูลฐาน ทำหน้าที่เป็นภาคประธาน ภาคแสดงและส่วนเสริม ทำหน้าที่เป็นส่วนขยาย

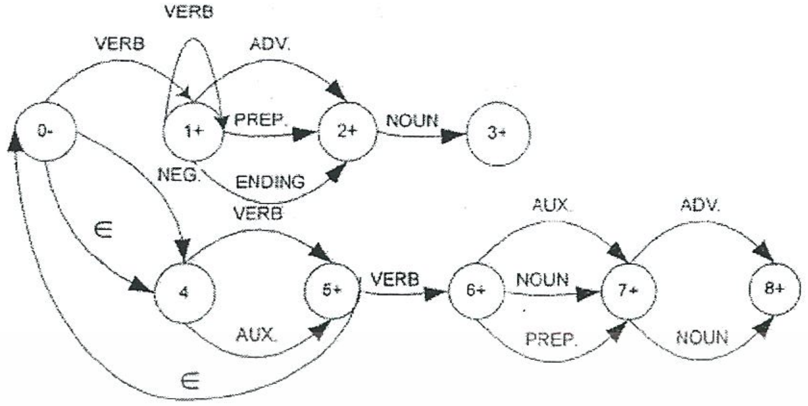
โครงสร้างวลีของภาษาไทยที่เป็นส่วนมูลฐาน

- นามวลี (noun phrase : NP)



ภาพที่ 2.3 : โครงสร้างวลีของนามวลี

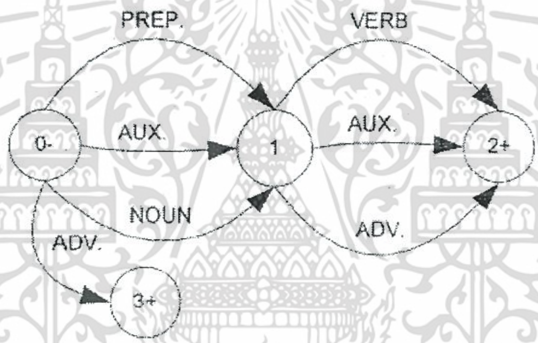
- กริยาวลี (Verb phrase : VP)



ภาพที่ 2.4 : โครงสร้างวลีของกริยาวลี

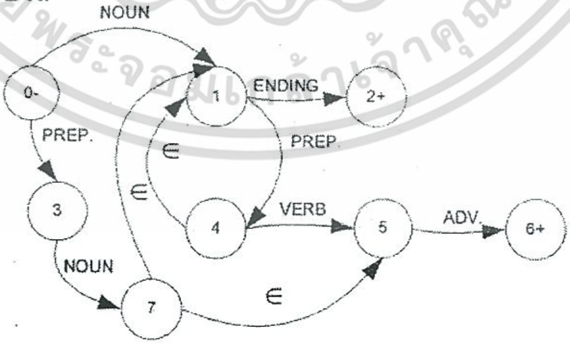
โครงสร้างวลีภาษาไทยที่เป็นส่วนเสริม

- สถานวิเศษณ์วลี



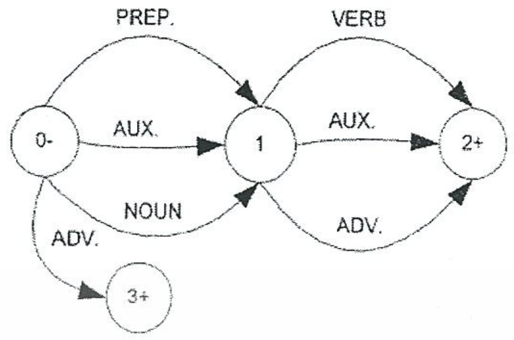
ภาพที่ 2.5 : โครงสร้างวลีของสถานวิเศษณ์วลี

- กาลวิเศษวลี



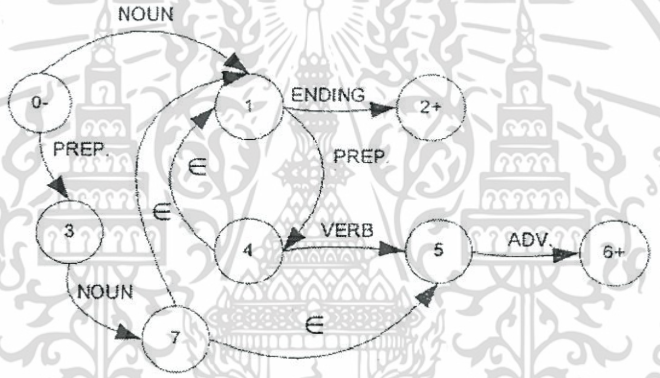
ภาพที่ 2.6 : โครงสร้างวลีของกาลวิเศษวลี

- พิเศษวิเศษณ์วลี



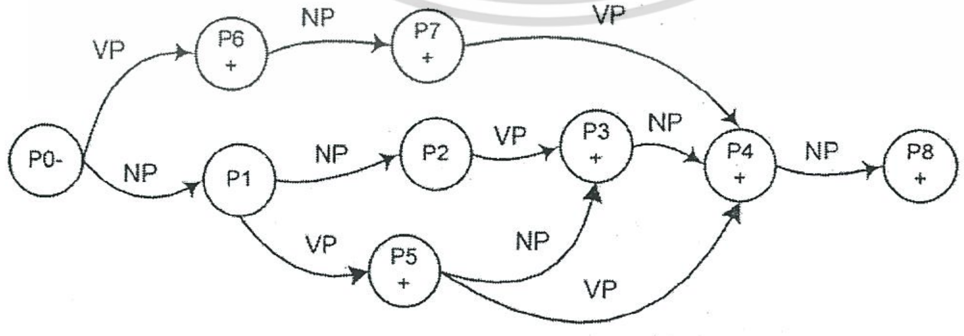
ภาพที่ 2.7 : โครงสร้างวลีของพิเศษวิเศษณ์วลี

- อาลปนะวลี



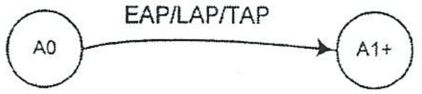
ภาพที่ 2.8 : โครงสร้างวลีของอาลปนะวลี

โครงสร้างประโยค ส่วนมูลฐานทำหน้าที่เป็นหน่วยประธาน หน่วยกรรม หน่วยนามเดี่ยว (NP) และทำหน้าที่เป็นหน่วยกริยา (VP) ข่ายงานความสัมพันธ์ของส่วนมูลฐานแสดงดังภาพที่ 2.9



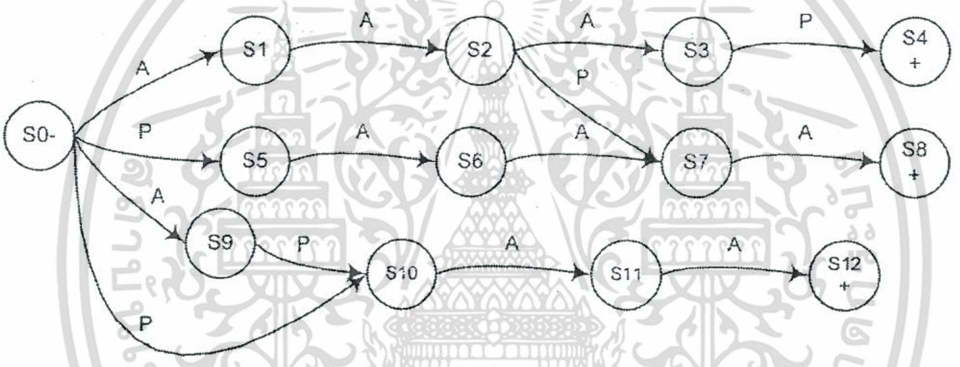
ภาพที่ 2.9 : ข่ายงานความสัมพันธ์ของส่วนมูลฐานในประโยค

ส่วนเสริมสามารถเรียงลำดับโดยเริ่มต้นจากวลีใดๆ และสลับตำแหน่งโดยไม่ทำให้ความหมายของประโยคเปลี่ยนแปลงจากภาพที่ 2.10 เป็นข่ายงานการเรียงลำดับของส่วนเสริม เมื่อ EAP คือพิเศษวิเศษณ์วลี , LAP คือ สถานวิเศษณ์วลี และ TAP คือ กาลวิเศษณ์วลี



ภาพที่ 2.10 : ข่ายงานความสัมพันธ์ของส่วนเสริมในประโยค

จากความสัมพันธ์ของส่วนมูลฐานและการเรียงลำดับของส่วนเสริม สามารถสร้างโครงสร้างประโยคภาษาไทยด้วย ATNs ได้ทั้งมด 4 โครงสร้าง โดยให้ P เป็นส่วนมูลฐานและให้ A เป็นส่วนเสริมดังข่ายงานที่แสดงในภาพที่ 2.11



ภาพที่ 2.11 : ข่ายงานความสัมพันธ์ของความสัมพันธ์ของส่วนมูลฐานและส่วนเสริม

ขั้นตอนการแปลประโยคโดยใช้ข่ายงานเพิ่มขยาย ประกอบไปด้วย การวิเคราะห์โครงสร้างวลีและโครงสร้างประโยค และการเลือกความหมายโดยพิจรณาจากบริบทที่เกิดร่วม

- การวิเคราะห์โครงสร้างวลีและโครงสร้างประโยค โดยใช้ข่ายงานเพิ่มขยาย Augmented Transition Network (ATNs)
- การเลือกคำแปลที่เหมาะสมพิจรณาจากบริบท และการเกิดร่วมของคำ จะใช้คุณสมบัติ relation ซึ่งเป็นตัวแปลคลาสจองแต่ละคำ เช่นคำว่า “ม่วน” มีหลายคำแปล แต่ละคำตัวแปลคลาส relation จะเก็บคำที่เกิดร่วมเช่น

คำว่าม่วน แปลว่า สนุก relation เล่น-เที่ยว

คำว่าม่วน แปลว่า เพราะ relation ฟัง-เพลง-พูด

ขั้นตอนการแปลเป็นภาษาไทยด้วยโครงสร้างไวยากรณ์ ตามหลัก CFG โดยมีกฎโครงสร้างวลี 5 กฎและกฎย่อยภายใต้โครงสร้างอีก 40 กฎ

2.2.3 การแปลภาษาระดับประโยค

ประโยค [3] เกิดจากกลุ่มคำหรือวลีประกอบกันเข้าเป็นประโยค โดยกลุ่มคำที่นำมาประกอบกันนั้นจะมีเนื้อความสมบูรณ์ บอกการกระทำความเป็นอยู่หรือความเป็นไปของสิ่งหนึ่งสิ่งใด การประกอบเข้ากันเป็นประโยคจะต้องประกอบด้วยภาคแสดงอย่างน้อยหนึ่งหน่วย เช่น “สุดา อ่านหนังสือ เมื่อวานนี้” ประโยคประกอบด้วย ภาคประธานและภาคแสดงโดย สุดา คือภาคประธานของประโยค และ อ่านหนังสือ เมื่อวานนี้ คือภาคแสดงในประโยค

ในการประมวลผลภาษาธรรมชาติ เพื่อให้คอมพิวเตอร์สามารถรับรู้และเข้าใจภาษามนุษย์ จำเป็นต้องนำหลักการทางภาษาศาสตร์ไปใช้ โครงสร้างประโยคของภาษานักภาษาศาสตร์จะมีการกำหนดกฎเกณฑ์มาช่วยตรวจวิเคราะห์และสร้างประโยค ตัวอย่างเช่น ไวยากรณ์ไม่พึ่งบริบท (Context Free Grammar) , ข่ายงานเพิ่มขยาย Augmented Transition Network (ATNs) เป็นต้น

ไวยากรณ์ไม่พึ่งบริบท (Context Free Grammar)[9] การแจงประโยคสำหรับไวยากรณ์ CFG อาจจะแจกแจงประโยคจากบนลงล่าง (Top-down parsing) และการแจงประโยคจากล่างขึ้นบน (Botton-up parsing) การแจงประโยคจากบนลงล่าง จะเริ่มที่สัญลักษณ์ประโยค S เขียนแทนด้วยสัญลักษณ์ นามวลี (NP) และตามด้วย กริยาวลี (VP) สัญลักษณ์ NP และ VP จะเขียนแทนต่อไปจนถึงสัญลักษณ์ จบท้ายที่แทนด้วยคำศัพท์ต่างๆ ลักษณะของการแจงประโยคแบบ บนลงล่างจะเป็นการใช้กฎไวยากรณ์ของสัญลักษณ์ ทางด้านขวาไปเขียนใหม่แทนสัญลักษณ์ทางด้านซ้าย ดังภาพที่ 2.12

S → NP + VP
→ Art + N + VP
→ The + N + VP
→ The birds + VP
→ The birds + V + NP
→ The birds eat + NP
→ The birds eat + Art + N
→ The birds eat the + N
→ The birds eat the worms.

ภาพที่ 2.12 : การแจงประโยคจากบนลงล่าง (Top-down parsing)

การแจงประโยคจากล่างขึ้นบน จะเริ่มต้นจากประโยค แล้วแทนด้วยคำศัพท์และชนิดของคำแทนสัญลักษณ์ของกฎทางขวาด้วยสัญลักษณ์ทางซ้ายไปจนถึงสัญลักษณ์ S ดังภาพที่ 2.13

→ The birds eat the worms.

→ Art + birds eat the worms.

→ Art + N + eat the worms.

→ Art + N + V + the worms.

→ Art + N + V + Art + worms.

→ <u>Art + N</u> + V + Art + N

→ NP + V + <u>Art + N</u>

→ NP + <u>V + NP</u>

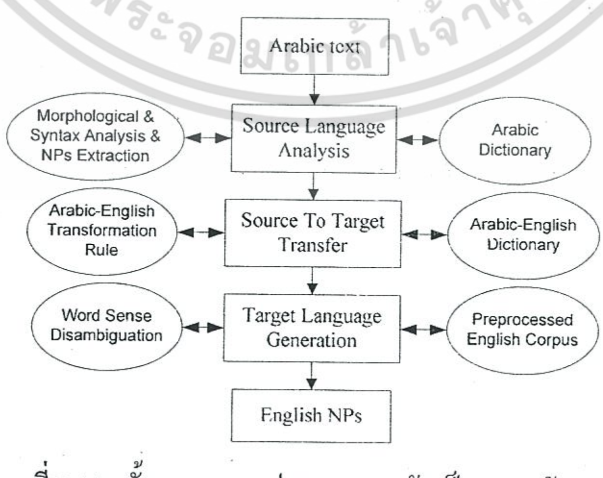
→ NP + VP

→ S

ภาพที่ 2.13 : การแจงประโยคจากล่างขึ้นบน (Botton-up parsing)

ในงานวิจัยการแปลภาษาอาหรับเป็นภาษาอักฤษ[6] งานวิจัยนี้ใช้วิเทอร์บีอัลกอริทึมในการเลือกตำแหน่งของโครงสร้างประโยคที่มีความเป็นไปได้สูงสุด จากคลังข้อมูลภาษาอังกฤษโดยสถาปัตยกรรมของการแปลภาษาอาหรับเป็นภาษาอังกฤษ ซึ่งแปลเฉพาะ Noun Phases ของประโยคอาหรับ

เป้าหมายหลักของระบบคือสร้างระบบการแปลโดยไม่ใช้ คลังคู่ขนานและสร้างกฎที่ครอบคลุม ภาพที่ 2.14 จะแสดงสถาปัตยกรรมของระบบแปลภาษาอาหรับเป็นภาษาอังกฤษ และทรัพยากรที่เกี่ยวข้องในแต่ละขั้นตอนการแปล ระบบแบ่งออกเป็น 3 ส่วนประกอบ คือ วิเคราะห์ประโยคต้นทาง (Source Language Analysis) , การถ่ายทอดจากภาษาต้นทางไปยังภาษาเป้าหมาย (Source to Target Transfer) และการสร้างประโยคภาษาเป้าหมาย (Target Language Generation)



ภาพที่ 2.14 : ขั้นตอนการแปลภาษาอาหรับเป็นภาษาอังกฤษ

ขั้นตอนที่ 1 วิเคราะห์ประโยคต้นทาง (Source Language Analysis) จะวิเคราะห์และแยกประโยคภาษาอาหรับ ในงานวิจัยดังกล่าวจะครอบคุลมเอกสารทางด้านการเกษตร หลังจากขั้นตอนการพาสซิ่งของประโยคนำเข้าภาษาอาหรับ นามวลีของภาษาอาหรับ (NP) จะถูกแยกโดยโครงสร้างไวยากรณ์ของภาษาอาหรับ

ขั้นตอนที่ 2 การถ่ายทอดจากภาษาต้นทางไปยังภาษาเป้าหมาย (Source to Target Transfer) จะทำการเปรียบเทียบโครงสร้างของประโยค ทำการเลือกโครงสร้างต้นไม้ของภาษา วิเคราะห์ต้นไม้ภาษาต้นทางเพื่อนำมาวิเคราะห์โครงสร้างต้นไม้ภาษาเป้าหมาย ด้านหนึ่งของต้นไม้ไวยากรณ์จะมีกฎของการถ่ายโอนโครงสร้างต้นไม้ ในการจับคู่กับโครงสร้างข้อมูลนำข้า (ภาษาอาหรับ) ผลลัพท์จะได้ต้นไม้ไวยากรณ์ทางด้านขวามือ

ในการแปล NP ที่นำเสนอ การแปลจะเกิดขึ้นในขั้นตอน transfer phase ในการแปลภาษาอาหรับเป็นภาษาอังกฤษ transfer phase มี 2 ขั้นตอน

- การถ่ายโอนคำศัพท์ (Lexical transfer) นอกจากจะเปรียบเทียบคำศัพท์ภาษาอาหรับเป็นภาษาอังกฤษแล้วยัง รวมไปถึงคุณสมบัติทางสัณฐานวิทยา (morphological) ที่ตรงกับคุณสมบัติภาษาอังกฤษ

- การถ่ายโอนโครงสร้าง (Structural transfer) จะเปรียบเทียบโครงสร้างต้นไม้ภาษาอาหรับ เทียบเท่ากับโครงสร้างของภาษาอังกฤษ

ขั้นตอนที่ 3 การสร้างประโยคภาษาเป้าหมาย (Target Language Generation) เป้าหมายคือสร้างภาษาเป้าหมาย โดยใช้คุณลักษณะของ สัณฐานวิทยา (morphological) แม้ในโมดูลก่อนหน้าจะลดความคุมเคลือในการแปลแต่ก็ยังคงมีความคลุมเคลือในระดับคำอยู่

ในโมดูลนี้ใช้วิธีการ dictionary-graph based WSD เป็นการตรวจสอบการทำงานร่วมกันของวิธีการ dictionary-based WSDและ graph-based WSD ทำให้การใช้คำที่ได้รับการแปลเป็นบริบท , บรรลุวัตถุประสงค์ขั้นพื้นฐาน WSD ในขั้นตอนของการแปล จะใช้พจนานุกรมอาหรับ-อังกฤษ ที่ได้อธิบายในข้างต้นแปลคำในแต่ละคำจาก ประโยคนำเข้าภาษาอาหรับ NP เพื่อแก้ปัญหาความคลุมเครือในการแปลภาษาอาหรับ NP ระบบการแปลระบุความคลุมเครือของคำและความสัมพันธ์ระหว่างกัน และใช้การค้นหาโดยอัลกอริทึมวิเทอร์บี (viterbi search) หาการแปลที่เหมาะสม ของคำภาษาอาหรับในการสร้างภาษาอังกฤษ NP

function VITERBI(*observations* of len *T*,*state-graph*) **returns** *best-path*

num-states ← NUM-OF-STATES(*state-graph*)
Create a path probability matrix *viterbi[num-states+2,T+2]*
viterbi[0,0] ← 1.0
for each time step *t* **from** 1 **to** *T* **do**
  for each state *s* **from** 1 **to** *num-states* **do**
    $viterbi[s,t] \leftarrow \max_{1 \le s' \le num\text{-}states} \left[viterbi[s', t-1] * a_{s',s} \right] * b_s(o_t)$
    $back\text{-}pointer[s,t] \leftarrow \operatorname{argmax}_{1 \le s' \le num\text{-}states} \left[viterbi[s', t-1] * a_{s',s} \right]$
Backtrace from highest probability state in final column of *viterbi[]* and return path

ภาพที่ 2.15 : ฟังก์ชั่นวิเทอร์บี สำหรับหาโครงสร้างประโยคภาษาอังกฤษ [6]

แนวคิดพื้นฐาน หากเรามีความคลุมเคลือของคำภาษาอาหรับ S ซึ่งมี 2 ประโยคคือ S1 และ S2 โดยที่ S1 จะสามารถแปลได้เป็น T1 และ S2 สามารถแปลได้เป็น T2 เพื่อที่จะทำให้เกิดความไม่คลุมเคลือในการเกิด S ในภาษาอาหรับ , เราจะระบุวลีที่เกิดขึ้นและใช้วิเทอร์บีเพื่อที่จะหาคำแปลที่เหมาะสมของคำภาษาอาหรับ S ขึ้นอยู่กับการสร้างภาษาเป้าหมายภาษาอังกฤษ NP

วิเทอร์บีอัลกอริทึม เป็นเทคนิคที่มีประสิทธิภาพ จะคำนวณลำดับโครงสร้างของประโยคภาษาเป้าหมายที่เป็นไปได้มากที่สุด จากงานวิจัยนี้ ทำให้ได้แนวคิดในการนำทฤษฎีของแบบจำลองฮิดเด้นมาร์คอฟ โดยใช้วิเทอร์บีอัลกอริทึม มาทำการลดความกำกวมของคำศัพท์ในการแปลภาษาไทยเป็นภาษาไทยล้านนา

ในงานวิจัยการแปลภาษาไทยล้านนาเป็นภาไทยนั้น[4] ได้ใช้แนวคิดว่าโครงสร้างของประโยคภาษาไทยล้านนาแปลเป็นภาษาไทยนั้นเป็นโครงสร้างประโยคเดียวกัน ซึ่งเมื่อแปลแล้วตำแหน่งของคำแต่ละคำในประโยคไทยล้านนา ก็จะอยู่ในตำแหน่งเดิมของประโยค ภาษาไทย แต่ในส่วนของโครงสร้างประโยคในการแปลจากภาษาไทยเป็นภาษาไทยล้านนานั้น พบว่ายังมีโครงสร้างประโยคบางประโยค ที่มีโครงสร้างที่ไม่ตรงตำแหน่งเดิมอยู่

ตัวอย่างของโครงสร้างประโยคไทยและโครงสร้างประโจคไทยล้านนา ที่โครงสร้างของประโยคมีความแตกต่างกัน

เขาวิ่งเร็วมาก เมื่อ แปลเป็นภาษาไทยล้านนาจะได้ “เขาล่นชาดเวย” ซึ่งคำใดๆ ของประโยคภาษาไทย ที่ทำหน้าที่เป็นกริยา เมื่อตามด้วยคำว่า มาก จะพบว่าสามารถแปลเป็นภาษาไทยล้านนาแล้วมีโครงสร้างของประโยคที่แตกต่างจากเดิม

ซึ่งในงานวิจัยนี้จะใช้วิธีการปรับปรุงตำแหน่งโครงสร้างประโยคด้วย ไวยากรณ์ปริวรรติเพิ่มพูน(Transformation Rule) ในการทำให้ในประโยคมีโครงสร้างคำที่มีความใกล้เคียงกับการที่มนุษย์มากยิ่งขึ้น

ตัววอย่างกฎไวยากรณ์ปริวรรตเพิ่มพูน ของภาษาไทยเป็นภาษาไทยล้านนา ประโยค "ส้มตำอร่อยมาก" ในภาษาไทยสามารถแปลเป็นภาษาไทยล้านนาได้มากกว่า 1 โครงสร้าง คือ "ตำส้มจ้าดลำ" และ "ตำส้มลำขนาด"

ส้มตำ/N อร่อย/V มาก/Adv

ตำส้ม/N จ้าด/Adv ลำ/V　　　ตำส้ม/N ลำ/V ขนาด/Adv

กฎการไวยากรณ์ปริวรรตของกริยาช่วยของคำว่า มาก (Adv) เมื่อไวยากรณ์ที่ผ่านเข้ามายังไวยากรณ์ปริวรรตเมื่อพบคำว่า "มาก" ที่มีคู่ภาษาล้านนาเป็น "จ้าด" ในภาษาไทยล้านนา จะปรับตำแหน่งประเภทของคำที่เป็นกริยาช่วย ไปอยู่ด้านหน้าของกริยา ดังต่อไปนี้

SD : NP V Adv

[มาก]

SC : NP Adv V

[จ้าด]

ผลลัพธ์ : จะได้โครงสร้างใหม่เพิ่มและโครงสร้างเดิม คือ

- ตำส้ม/N จ้าด/Adv ลำ/V
- ตำส้ม/N ลำ/V *ขนาด*/Adv

ดังนั้น เมื่อผ่านไวยากรณ์ปริวรรติเพิ่มพูน ประโยคไทยล้านนาดังกล่าวจะทำให้ผลลัพท์ของการแปลภาษาไทยเป็นภาษาไทยล้านนมีโครงสร้างประโยคเมื่อปรับปรุงโครงสร้างแล้วมีความใกล้เคียงกับการใช้งานของมนุษย์มากขึ้น

บทที่ 3
วิธีการดำเนินงานวิจัย

งานวิจัยนี้มีวัตถุประสงค์ในการแปลภาษาไทยเป็นภาษาล้านนา โดยเลือกความหมายจากพจนานุกรมไทยล้านนา วิเคราะห์โครงสร้างประโยคภาษาไทยด้วยไวยากรณ์ไม่พึ่งบริบทประยุกต์เทคนิคทางสถิติ (Probabilistic Context free grammar : PCFG) ปรับโครงสร้างไวยากรณ์ให้เป็นโครงสร้างไวยากรณ์ไทยล้านนาด้วย ไวยากรณ์ปริวรรต (Transformation Grammar) และลดความกำกวมด้วยวิธีการของแบบจำลองฮิดเด้นมาร์คอฟและวิเทอร์บิอัลกอริทึม (Viterbi algorithm) ผลลัพธ์จะได้ประโยคภาษาล้านนาซึ่งเป็นภาษาเป้าหมาย

3.1 สถาปัตยกรรมของการแปลภาษาไทยเป็นภาษาไทยล้านนา

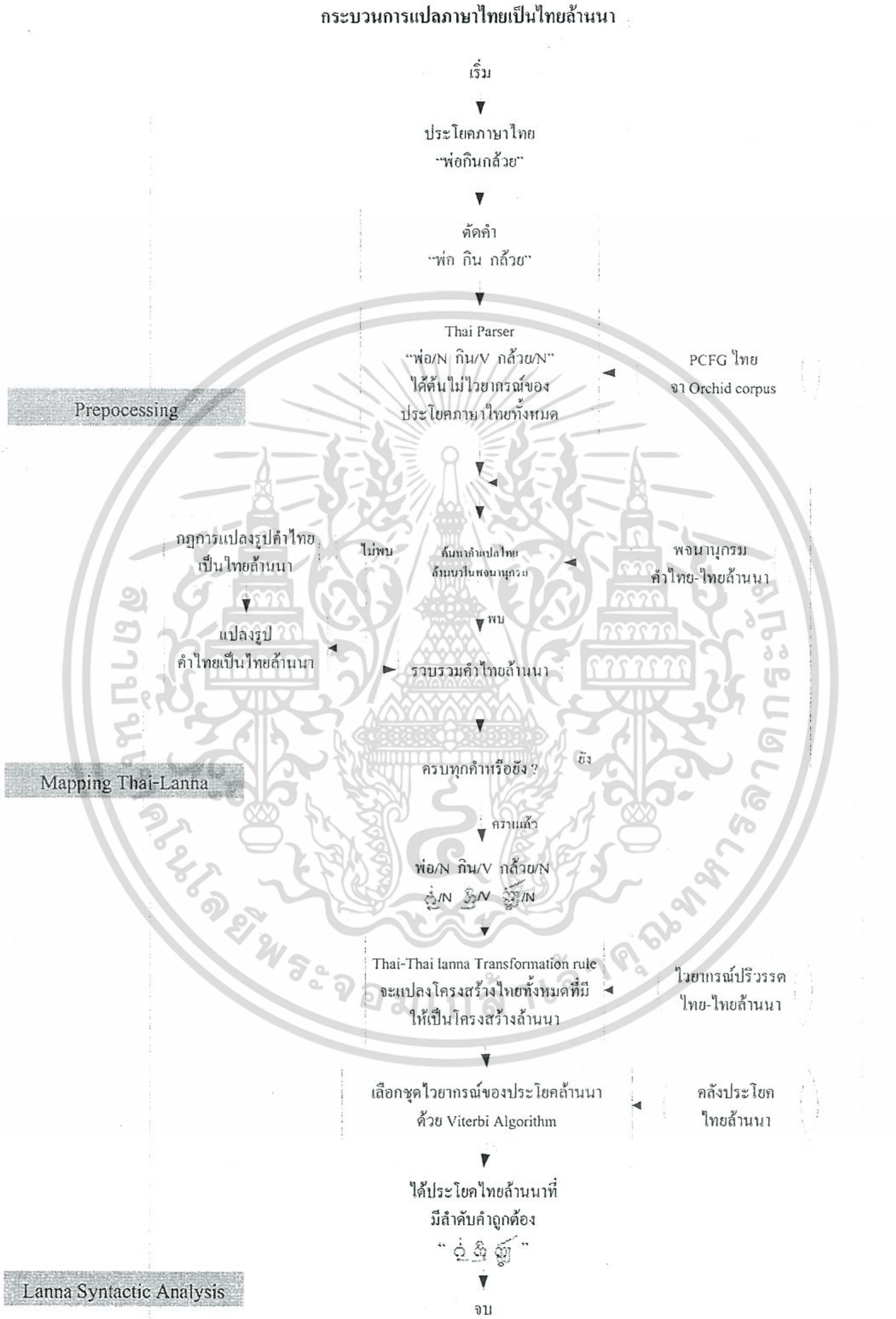
สถาปัตยกรรมของแปลภาษาไทยเป็นภาษาไทยล้านนาได้แบ่งขั้นตอนการทำงานออก เป็นขั้นตอนดังภาพที่ 3.1 ซึ่งประกอบด้วยขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 หลังจากที่ผู้ใช้ใส่ประโยคภาษาไทยที่ต้องการแปล จะเข้าสู่ขั้นตอนก่อนการประมวลผล (Preprocessing) ซึ่งในขั้นตอนนี้นั้น จะเป็นขั้นตอนของการเตรียมข้อมูลก่อนการแปล จะเป็นการตัดคำภาษาไทยโดยใช้เครื่องมือตัดคำ (KU-CUT) ซึ่งเป็นเครื่องมือที่ตัดคำด้วยเทคนิคการเรียนรู้จากข้อมูลแบบ ไม่ใช้ตัวอย่างร่วมกับการใช้พจนานุกรมและคำ จากนั้นจะนำข้อมูลที่ผ่านการตัดคำนั้นๆ ไปเข้าสู่การวิเคราะห์โครงสร้าง ผ่านเครื่องจักรวิเคราะห์ประโยคโดยใช้ไวยากรณ์ไม่พึ่งบริบทประยุกต์เทคนิคทางสถิติ (Probalistic Context free grammar : PCFG) เพื่อวิเคราะห์และจัดรูปแบบตำแหน่งของประเภทของคำ ให้ตรงกับโครงสร้างประโยคไทย ผลลัพธ์จะได้โครสร้างประโยคภาษาไทยที่ถูกต้องและมีการกำกับ ชนิดของคำ (Part of speech : POS) จากประโยคตั้งต้น

ขั้นตอนที่ 2 การแปลประโยคไทยล้านนา (Mapping Thai-Lanna) ในขั้นตอนนี้จะทำการนำคำศัพท์ภาษาไทยในขั้นตอนที่ 1 ไปค้นหาคำแปลภาษาไทยล้านนาจากพจนานุกรมไทย-ล้านนาซึ่งมีการจัดเก็บข้อมูลในโครงสร้างแบบทรัย (Trie) หากในกรณีที่ไม่พบคำแปลในพจนานุกรมจะใช้วิธีการแปลงรูปคำไทยล้านนาตามกฎของโครงสร้างของการเขียนภาษาไทยล้านนาและระบุชนิดของคำนั้นเป็นคำนามชนิดวิสามานยนาม ผลลัพธ์ในขั้นตอนนี้ก็จะได้คำแปลล้านนากำกับอยู่ในแต่ละคำไทย

ขั้นตอนที่ 3 การสร้างประโยคล้านนา (Lanna Syntactic Analysis) คำภาษาไทย ที่มีการแจกแจงโครงสร้างประโยค และกำกับชนิดของคำไทย จะนำเข้าสู่ไวยากรณ์ปริวรรตโดยจะมีกฎของการปรับให้เป็นโครงสร้างประโยคไทยล้านนา จากนั้นจะลดความกำกวมของโครงสร้างประโยคและลำดับของคำล้านนา โดยใช้แบบจำลองฮิดเด้นมาร์คอฟ วิธีการวิเทอร์บิอัลกอริทึม โดย

คำนวณสถิติจากคลังภาษาไทยล้านนา โดย ผลลัพธ์ก็จะได้ประโยคไทยล้านนาที่โครงสร้างและลำดับคำที่ถูกต้อง



ภาพที่ 3.1 : สถาปัตยกรรมของการแปลภาษาไทยเป็นภาษาไทยล้านนา

3.2 ไวยากรณ์ไม่พึ่งบริบทใช้เทคนิคทางสถิติ (Probabilistic Context Free Grammar : PCFG)

ไวยากรณ์ไม่พึ่งบริบท [9] เป็นส่วนหนึ่งของไวยากรณ์โครงสร้างวลี เพื่อใช้อธิบายโครงสร้างของภาษาและความสัมพันธ์ระหว่างโครงสร้างต่างๆ ที่ประกอบในประโยคนั้นๆ ถูกนิยามโดย

G = (V, T, S, P)

โดยที่ V คือ เซตของตัวแปล (Variable)

T คือ เซตของสัญลักษณ์เทอร์มินอล (Terminal symbol)

S คือ สัญลักษณ์เริ่มต้น (Start symbol)

P คือ เซตของโปรดักชั่น (Production) ซึ่งมีรูปแบบดังนี้ A -> a

โดยที่ A คือ ตัวแปล (A € V) และ *a* คือสตริงของ (V Union T)

ในงานวิจัยนี้กฎโครงสร้างวลีและโครงสร้างประโยคของภาษาไทย อ้างอิงจากบทความ การวิเคราะห์ประโยคภาษาไทย ของเรืองเดช ปันเขื่อนขัติย์ [7] ซึ่งสามารถสร้างกฎโครงสร้างวลีจำนวน 5 กฎ และกฎย่อยภายใต้โครงสร้างวลีอีก 79 กฎ สามารถแยกตามกฎโครงสร้างวลี เพื่อนำมาสร้างเป็นกฎโครงสร้างประโยคภาษาไทยดังตาราง

โครงสร้างนามวลี (NP) แบ่งออกเป็น 13 กฎ	
1) NOUN	8) NOUN + CLASS + NOUN
2) ADJ	9) NOUN + CLASS + CLASS
3) NOUN + CLASS	10) NOUN + CLASS + DETER
4) NOUN + NOUN +CLASS	11) ADJ + NOUN + NOUN
5) NOUN + ADJ	12) ADJ + NOUN + CLASS
6) NOUN + NOUN	13) ADJ + NOUN + DETER
7) NOUN +DETER	

ตารางที่ 3.1 : ตารางโครงสร้างนามวลี

โครงสร้างกริยาวลี (VP) แบ่งออกเป็น 41 กฎ	
1) VERB	22) AUX + VERB + NOUN
2) VERB + VERB	23) AUX + VERB + PREP
3) VERB + ADV	24) NEG + VERB + VERB + AUX + ADV
4) VERB + PREP	25) NEG + VERB + VERB + AUX + NOUN
5) VERB + ENDING	26) NEG + VERB + VERB + NOUN + ADV
6) VERB + ADV + NOUN	27) NEG + VERB + VERB + NOUN + NOUN

7) VERB + PREP + NOUN	28) NEG + VERB + VERB + PREP + ADV
8) VERB + ENDING + NOUN	29) NEG + VERB + VERB + PREP + NOUN
9) NEG + VERB	30) NEG + AUX +VERB + AUX + ADV
10) NEG + AUX	31) NEG + AUX +VERB + AUX + NOUN
11) AUX	32) NEG + AUX +VERB + NOUN + ADV
12) NEG + VERB + VERB	33) NEG + AUX +VERB + NOUN + NOUN
13) NEG + AUX +VERB	34) NEG + AUX +VERB + PREP + ADV
14) AUX + VERB	35) NEG + AUX +VERB + PREP + NOUN
15) NEG + VERB + VERB + AUX	36) AUX + VERB + AUX + ADV
16) NEG + VERB + VERB + NOUN	37) AUX + VERB + AUX + NOUN
17) NEG + VERB + VERB + PREP	38) AUX + VERB + NOUN + ADV
18) NEG + AUX +VERB + AUX	39) AUX + VERB + NOUN + NOUN
19) NEG + AUX +VERB + NOUN	40) AUX + VERB + PREP + ADV
20) NEG + AUX +VERB + PREP	41) AUX + VERB + PREP + NOUN
21) AUX + VERB + AUX	

ตารางที่ 3.2 :ตารางโครงสร้างกริยาวลี

โครงสร้างพิเศษวลี (EAP) แบ่งออกเป็น 10 กฎ	
1) PREP + VERB	6) AUX + ADV
2) PREP + AUX	7) NOUN + VERB
3) PREP + ADV	8) NOUN + AUX
4) AUX + VERB	9) NOUN + ADV
5) AUX + AUX	10) ADV

ตารางที่ 3.3 :ตารางโครงสร้างพิเศษวลี

โครงสร้างกาลวิเศษณ์วลี (TAP) แบ่งออกเป็น 5 กฎ	
1) NOUN + ENDING	4) PREP + NOUN + ENDING
2) NOUN + PREP + VERB + ADV	5) PREP + NOUN + PREP + VERB + ADV
3) PREP + NOUN + ADV	

ตารางที่ 3.4 :ตารางโครงสร้างกาลวิเศษณ์วลี

โครงสร้างสถานวิเศษณ์วลี (TAP) แบ่งออกเป็น 10 กฎ	
1) PREP + VERB	6) AUX + ADV
2) PREP + AUX	7) NOUN + VERB
3) PREP + ADV	8) NOUN + AUX
4) AUX + VERB	9) NOUN + ADV
5) AUX + AUX	10) ADV

ตารางที่ 3.5 : โครงสร้างสถานวิเศษณ์วลี

จากแจงประโยคโดยไวยากรณ์ไม่พึ่งบริบทจะได้ผลลัพธ์ออกมาเป็นต้นไม้ไวยากรณ์ที่มีได้หลายโครงสร้าง ดังนั้นในงานวิจัยนี้จึงใช้เทคนิคทางสถิติ (Probabilistic Context Free Grammar : PCFG) มาช่วยในการวิเคราะห์โครงสร้างไวยากรณ์ จะมีการกำหนดค่าความน่าจะเป็นสำหรับแต่ละกฎไวยากรณ์ ซึ่งมีสมมุติฐานว่า แต่ละไวยากรณ์มีการเกิดขึ้นแบบอิสระต่อกัน โดยโอกาสเกิดของกฎแต่ละวลีนั้นมีโอกาสเกิดขึ้นเป็น 1

ค่าสถิติของแต่ละกฎสามารถหาได้จากการนับการเกิดขึ้นของกฎนั้นๆ เทียบกับกฎอื่นๆ ที่มีวลีนั้นๆ เป็นชนิดเดียวกันจากคลังประโยคของ Orchid Corpus ซึ่งเป็นคลังข้อมูลประโยคภาษาไทยของศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ มีการกำกับชนิดของคำและมีโครงสร้างของประโยค ผลลัพธ์จะทำให้ได้โครงสร้างไวยากรณ์ภาษาไทยที่มีค่าความน่าจะเป็นสูงสุด 1 โครงสร้าง

3.3 โครงสร้างพจนานุกรมไทย-ล้านนา

พจนานุกรมคู่ภาษาไทย-ไทยล้านนาสร้างขึ้นจาก พจนานุกรมล้านนา-ไทย ฉบับเฉลิมพระเกียรติพระบาท สมเด็จพระเจ้าอยู่หัว ในวโรกาสเจริญพระชนมายุ 80 พรรษา ของสถาบันภาษาและศิลปวัฒนธรรม มหาวิทยาลัยราชภัฏเชียงใหม่ รวบรวมข้อมูลคู่คำศัพท์ทั้งหมด 19,650 คำ โดยจะจัดเก็บโครงสร้างข้อมูลแบบทรัย (Trie)

โครงสร้างข้อมูลแบบทรัย (Trie) มีลักษณะคล้ายกับโครงสร้างข้อมูลแบบต้นไม้แต่วิธีการจัดเก็บข้อมูลจะแตกต่างกัน โดยที่โครงสร้างข้อมูลแบบทรัยนี้จะจัดเก็บตัวอักขระของคำศัพท์ซึ่งโครงสร้างข้อมูลแบบต้นไม้นั้นจะจัดเก็บข้อมูลทั้งคำ โดยโครงสร้างของทรัยจะประกอบไปด้วยโหนดต่างๆ จะประกอบไปด้วยดัชนีที่ชี้ไปยังโหนดของตัวอักขระถัดไปซึ่งมีจำนวนดัชนีเท่ากับค่าของจำนวนตัวอักขระที่จะอนุญาตให้มีได้ในพจนานุกรมบวกกับอักขระที่ใช้ระบุเป็นตัวจบคำศัพท์ (Terminator)

สำหรับการสืบค้นในโครงสร้างข้อมูลแบบทรัยนี้จะทำโดยเริ่มต้นที่โหนดศูนย์ถ้าต้องการค้นหาคำศัพท์ก็ให้นำอักขระทีละตัวจากคำศัพท์ที่ต้องการมาดูว่าภายในโหนด 0 นั้นมีดัชนีของตัวอักขระที่ต้องการไปชี้โหนดอื่นหรือไม่ ถ้าไม่มีแสดงว่าคำนั้นไม่มีอยู่ในพจนานุกรม แต่ถ้ามีดัชนี

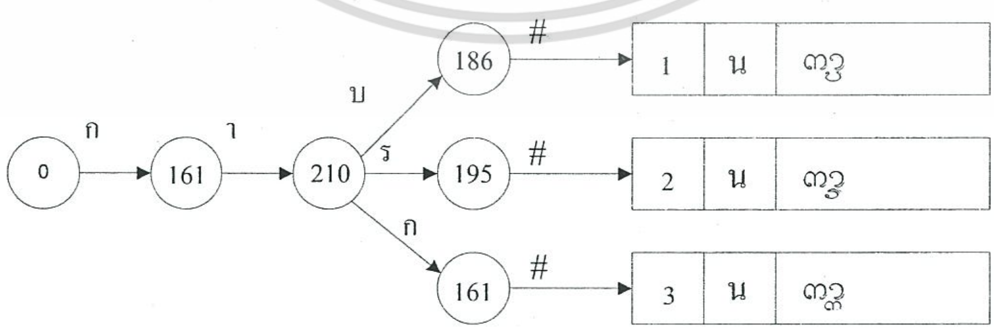
ที่ชี้ไปโหนดถัดไปก็ให้เดินที่โหนดที่ดัชนีนั้นชี้ไปแล้วนำตัวอักขระตัวถัดไปมาทำตามขั้นตอนแบบเดิมจนหมด เมื่อนำตัวอักขระทั้งหมดจากดัชนีมาเดินในทรัยแล้วให้สิ้นอักขระ ถ้าดัชนีมีค่าเท่ากับค่าว่าง (Null) แสดงว่าไม่มีคำศัพท์นั้นในพจนานุกรม แต่ถ้าไม่เท่ากับค่าว่างแสดงว่ามีคำศัพท์นั้นอยู่ในพจนานุกรม โดยดัชนีเป็นตัวชี้ตำแหน่งของข้อมูลของคำนั้น ส่วนความเร็วในการค้นหาภายในโครงสร้างข้อมูลแบบทรัยนั้นจะไม่ขึ้นอยู่กับจำนวนคำที่มีในพจนานุกรม แต่จะขึ้นอยู่กับจำนวนดัชนีของตัวอักษร

ประเภทของคำในพจนานุกรม ประเภทของคำแบ่งตามหลักภาษาศาสตร์แบ่งออกเป็น 11 ประเภทดังตารางที่ 3.6

ประเภทคำ	ความหมาย
Noun	คำนาม
Pronoun	คำสรรพนาม
Verb	คำกริยา
Auxillary	คำวิเศษณ์
Prepposition	คำบุพบท
Conjunction	คำสันธาน
Ending	คำลงท้าย
Interjunction	คำอุทาน
Prefix	หน่วยคำเติมหน้า
Negation	ปฏิเสธ

ตารางที่ 3.6 :ประเภทของคำ

จากภาพที่ 3.2 เป็นตัวอย่างการจัดเก็บข้อมูลในพจนานุกรมไทย-ไทยล้านนา ด้วยโครงสร้างข้อมูลแบบทรัย



ภาพที่ 3.2 : ตัวอย่างโครงสร้างข้อมูลของพจนานุกรมไทย-ไทยล้านนา

3.4 การแปลงรูปภาษาไทยล้านนา

การแปลงรูปคำไทยเป็นไทยล้านนา ในขั้นตอนการค้นหาคำศัพท์ในพจนานุกรมจะนำคำภาษาไทยที่ได้จากการตัดคำไปค้นหาคำแปลที่เป็นคู่คำศัพท์ จากพจนานุกรมคู่ภาษาไทย-ภาษาไทยล้านนา หากคำภาษาไทยไม่สามารถจับคู่ได้ จะแปลงคำไทยนั้นเป็นคำไทยล้านนาด้วยกฎการเรียงอักษรไทยล้านนาที่ได้มาจากผู้เชี่ยวชาญและการแปลงรูปภาษาไทยล้านนา

การปริวรรตอักษรธรรมล้านนาเป็นอักษรไทยกลางดังกล่าว อาจทำได้ 3 ระดับ คือ

1. การปริวรรตโดยการถ่ายเทียบตัวอักษรและเครื่องหมายทุกชนิด ที่ปรากฏในต้นฉบับโดยเคร่งครัด
2. การปริวรรตโดยใช้อักษรและวิธีผสมอักษรแบบไทยกลาง
3. การปริวรรตโดยใช้อักษรไทยกลางบันทึกคำอ่านในภาษาถิ่นล้านนา

ที่ประชุมข้อตกลงในการปริวรรตอักษรธรรมล้านนาเป็นอักษรไทยกลาง มติที่ประชุมเชิงปฏิบัติการเรื่องมาตรฐาน การปริวรรตอักษรพื้นเมืองล้านนาโดย โครงการศูนย์ส่งเสริมศิลปวัฒนธรรม มหาวิทยาลัยเชียงใหม่ ได้สรุปว่า วิธีการปริวรรตตามแบบแรกนั้น เหมาะสำหรับการปริวรรตเอกสารที่ต้องการความถี่ถ้วนเป็นพิเศษ เช่น จารึกชนิดต่างๆ การปริวรรตตามแบบที่สองเหมาะสำหรับการเผยแพร่แก่สาธารชนผู้สนใจวรรณกรรมล้านนาโดยทั่วไป ส่วนการปริวรรตตามแบบสุดท้ายนั้น เหมาะที่จะเผยแพร่แก่ประชาชนในท้องถิ่นที่พูดภาษาล้านนาโดยเฉพาะ

กฎที่สร้างขึ้นพิจารณาลำดับของการประสมคำไทยล้านนาด้วยสระต่างๆ เป็นหลักโดยกำหนดสัญลักษณ์ต่างๆ ที่ใช้ในการสร้างกฎดังนี้

C แทนพยัญชนะต้นหรือพยัญชนะท้าย ในกรณีที่ไม่ทราบแน่นอนว่าพยัญชนะนั้นเป็นตัวสะกดหรือพยัญชนะท้าย

T แทนวรรณยุกต์

V แทนสระ

S แทนพยัญชนะท้ายหรือตัวสะกดในกรณีที่ทราบแน่นอนว่าพยัญชนะนั้นเป็นตัวสะกด หรือพยัญชนะต้น

$[a_1, a_2, a_3, \ldots, a_n]$ แทนการเลือกเอาตัวใดตัวหนึ่งระหว่าง $a_1, a_2, a_3, \ldots, a_n$

เนื่องจากภาษาไทยและภาษาล้านนามีตัวควบกล้ำ ซึ่งหากใช้ CC แทนพยัญชนะควบกล้ำแล้วจะทำให้กำกวมได้เพราะในภาษาล้านนา พยัญชนะบางตัวทำหน้าที่เป็นสระด้วย ดังนั้นจึงเพิ่มสัญลักษณ์ควบกล้ำเพื่อลดความกำกวมดังกล่าว

CR แทนพยัญชนะควบกล้ำ ร

CL แทนพยัญชนะควบกล้ำ ล

CW แทนพยัญชนะควบกล้ำ ว

HC แทนพยัญชนะ ห นำพยัญชนะต้น

AI แทนพยัญชนะ อย

O แทนสระ ออ ที่มีพยัญชนะท้าย

Y แทนสระเอีย

กฎที่	กฎ	โครงสร้าง	กฎที่	กฎ	โครงสร้าง
1	[CV]ะ	CTVS	17	[CT]ัวะ	CTV
2	[CV]า	CTVS	18	[CT]ัว	CTV
3	[CV]ิ	CTVS	19	เ[CT]ียะ	VCTV
4	[CV]ี	CTVS	20	เ[CT]ีย	VCTV
5	[CV]ึ	CTVS	21	เ[CT]อะ	VCTV
6	[CV]ื	CTVS	22	เ[CT]ือ	VCTV
7	[CV]ุ	CTVS	23	เ[CT]อะ	VCTV
8	[CV]ู	CTVS	24	เ[CT]อ	VCTV
9	เ[CT]ะ	VCTV	25	ไ[CT]	VCTS
10	เ[CT]	VCT	26	ใ[CT]	VCTS
11	แ[CT]ะ	VCTV	27	ไ[CT]ย	VCTV
12	แ[CT]	VCT	28	ไ[CT]ยย	VCTV
13	โ[CT]ะ	VCTV	29	เ[CT]า	VCTV
14	โ[CT]	VCT	30	[CT]ำ	CTV
15	เ[CT]าะ	VCTV	31	[CT]ํ	CTV
16	[CT]อ	CTV			

ตารางที่ 3.7 : กฎการแปลงรูปภาษาไทยเป็นภาษาล้านนาที่ได้สร้างขึ้น

ในการพิมพ์อักขระภาษาไทยล้านนาในคอมพิวเตอร์ ในอดีตนั้นจะใช้วิธีการพิมพ์ภาษาอังกฤษเข้าไปแทนที่อักขระพิเศษเช่น คำว่า *"กิน"* ซึ่งมี *"น"* เป็นตัวสะกด ในการพิมพ์ภาษาไทยล้านนาตัวสะกดจะต้องอยู่ใต้คำ ดังนั้นจะต้องเขียนเป็น *"กิo"* ทำให้ผู้ใช้ต้องสลับโหมดระหว่างภาษาไทยกับภาษาอังกฤษบ่อยครั้ง การพิมพ์จึงล่าช้า และไม่สะดวกเท่าที่ควร ด้วยเหตุนี้ ในปี พ.ศ.2549 สำนักส่งเสริมศิลปะวัฒนธรรม มหาวิทยาลัยเชียงใหม่ จึงได้ทำโครงการ "พัฒนาระบบการพิมพ์อักษรธรรมล้านนา" และโครงการ "พัฒนาแม่แบบชุดอักษร *Lanna OTF Template"* ขึ้น เพื่อแก้ปัญหาระบบการพิมพ์ดังกล่าว โดยพัฒนาฟอนต์ Ln-tilok และระบบการพิมพ์ให้ใช้ง่าย และมีผลกระทบกับความเคยชิน ในระบบพิมพ์สัมผัสของผู้ใช้ให้น้อยที่สุด ทั้งนี้

ได้ศึกษาค้น คว้าหาวิธีที่จะทำให้การพิมพ์อักษรธรรมล้านนานั้น อยู่ภายในแป้นพิมพ์เดียว พัฒนามาจนถึงรุ่น Ln-tilok 6.00 ในปี พ.ศ.2552

ในงานวิจัยการแปลภาษาไทยเป็นภาษาไทยล้านนาในการเขียนภาษาไทยล้านนา ในคอมพิวเตอร์นั้นได้อ้างอิงวิธีการพิมพ์ตัวอักขระภาษาไทยล้านนาโดยใช้ฟอน์ต Ln-tilok ในการพิมพ์รูปแบบตัวอักษรไทยล้านนา ซึ่งในการพิมพ์จะใช้อักขระ "ฯ" และ "฿" มาช่วยในการพิมพ์ เพราะตัวอักขระสองตัวนี้ไม่พบในภาษาไทยล้านนาโดยเมื่อใดก็ตามที่ต้องการพิมพ์ตัวสะกดเช่น

กิน	เขียนเป็นภาษาไทยล้านนาได้	กินฯ
ทาน	เขียนเป็นภาษาไทยล้านนาได้	ทานฯ
การ	เขียนเป็นภาษาไทยล้านนาได้	การฯ
วัด	เขียนเป็นภาษาไทยล้านนาได้	วัดฯ
ชาด	เขียนเป็นภาษาไทยล้านนาได้	ชาดฯ

ส่วนสัญลักษณ์ "฿" จะใช้สำหรับสระโอะลดรูป ตัวอย่างเช่น

มด	เขียนเป็นภาษาไทยล้านนาได้	ม฿ด
ลด	เขียนเป็นภาษาไทยล้านนาได้	ล฿ด
จด	เขียนเป็นภาษาไทยล้านนาได้	จ฿ด

3.5 ไวยากรณ์ปริวรรติเพิ่มพูน (Transformation generative grammar)

ชอมสกี้ [9] ได้กล่าวถึงไวยากรณ์โครงสร้างวลีและแสดงให้เห็นว่าหลักไวยากรณ์นี้ยังไม่เพียงพอในการอธิบายภาษาธรรมชาติ จึงได้พัฒนาไวยากรณ์ปริวรรติเพิ่มพูน (Transformation generative grammar) ประโยคที่ผ่านไวยากรณ์ CFG แล้วจะให้ประโยคที่เรียกว่าโครงสร้างลึก โดยปกติได้อยู่ในรูปโครงสร้างต้นไม้ แต่โดยปกติประโยคที่เราใช้ในการพูดเรียกว่าโครงสร้างผิว

กฎการเปลี่ยนโครงสร้างวลีให้เป็นโครงสร้างผิวโดยการเพิ่มหรือลดตำแหน่งของคำซึ่งอาจจะแบ่งได้เป็นกฎบังคับที่ใช้ในการเปลี่ยนรูปโครงสร้างเพราะถ้าไม่เปลี่ยนจะทำให้ประโยคผิดหลักไวยากรณ์ อีกประเภทเป็นกฎที่ใช้หรือไม่ใช้ก็ได้ ประโยคจะไม่เปลี่ยนแปลงตามกฎก็ได้

กฎการปริวรรตจะแบ่งออกเป็น 3 ส่วนคือ

- ส่วนอธิบายโครงสร้าง (Structural description : SD) เพื่อแสดงโครงสร้างของประโยคว่าควรเป็นอย่างไรถึงใช้กฎนี้

- ส่วนแสดงการเปลี่ยนแปลง (Structural Change :SC) เพื่อแสดงโครงสร้างของประโยคที่เปลี่ยนแปลงไปหลังจากที่ปริวรรตแล้ว

- ส่วนของชุดเงื่อนไขที่ต้องทำการเปรียบเทียบให้ตรงกัน

ตัวอย่างกฎไวยากรณ์ปริวรรตเพิ่มพูน ของภาษาไทยเป็นภาษาไทยล้านนา ประโยค "ส้มตำอร่อยมาก" ในภาษาไทยสามารถแปลเป็นภาษาไทยล้านนาได้มากกว่า 1 โครงสร้าง คือ "ตำส้มจ้าดลำ" และ "ตำส้มลำขนาด"

ส้มตำ/N อร่อย/V มาก/Adv

ตำส้ม/N จ้าด/Adv ลำ/V　　ตำส้ม/N ลำ/V ขนาด/Adv

กฎการไวยากรณ์ปริวรรตของกริยาช่วยของคำว่า มาก (Adv) เมื่อไวยากรณ์ที่ผ่านเข้ามายังไวยากรณ์ปริวรรตเมื่อพบคำว่า “มาก” ที่มีคู่ภาษาล้านนาเป็น “จ้าด” ในภาษาไทยล้านนา จะปรับตำแหน่งประเภทของคำที่เป็นกริยาช่วย ไปอยู่ด้านหน้าของกริยา ดังต่อไปนี้

SD :	NP	V	Adv
			[มาก]
SC :	NP	Adv	V
		[จ้าด]	

ผลลัพธ์ : จะได้โครงสร้างใหม่เพิ่มและโครงสร้างเดิม คือ

- ตำส้ม/N จ้าด/Adv ลำ/V
- ตำส้ม/N ลำ/V ขนาด/Adv

3.6 แบบจำลองฮิดเดนมาร์คอฟและการลดความกำกวมด้วยวิเทอร์บีอัลกอริทึม

แบบจำลองฮิดเดนมาร์คอฟ [10] เป็นเทคนิคการเรียนรู้ทางสถิติ (Statistical Machine Learning) ที่ใช้จำแนกรูปแบบของลำดับเหตุการณ์หนึ่งๆ ซึ่งลำดับของเหตุการณ์จะถูกจำลองให้อยู่ในรูปแบบของลำดับการเปลี่ยนแปลงของสถานะ (State) จะมีค่าความน่าจะเป็นอยู่ 2 ชนิด คือ ค่าความน่าจะเป็นของการเปลี่ยนสถานะ และค่าความน่าจะเป็นของการทำให้เกิดผลลัพธ์ เมื่อมีการเปลี่ยนสถานะเกิดขึ้นและผลลัพธ์นี้จะถูกนำไปใช้ในการตัดสินใจว่าข้อมูลที่เข้ามามีโอกาสเป็นอะไรมากที่สุด

วิเทอร์บีอัลกอริทึมเป็นอัลกอริทึมหนึ่งในแบบจำลองฮิดเดนมาร์คอฟ ที่ใช้ในการจัดตำแหน่งลำดับของถานะต่างๆ โดยลำดับของแต่ละสถานะจะขึ้นอยู่กับความน่าจะเป็นสูงสุด ที่จะเป็นไปได้ทั้งหมด

การทำงานของวิเทอร์บีอัลกอริทึม มี 3 ขั้นตอน คือ

ขั้นตอนที่ 1 คำนวณค่าความน่าจะเป็นของสถานะเริ่มต้นของข้อมูลนำเข้า

ขั้นตอนที่ 2 คำนวณความน่าจะเป็นของสถานะถัดไป ทุกสถานะ เพื่อจะเก็บเส้นทางที่ดีที่สุด สำหรับการเลือกเส้นทางในขั้นตอนถัดไป

ขั้นตอนที่ 3 การย้อนเส้นทาง (Backtracking) เพื่อค้นหาลำดับความน่าจะเป็นสูงสุด จากสถานะที่ได้คำนวณไว้แล้ว ซึ่งจะได้ลำดับเส้นทางสถานะที่เป็นไปได้สูงสุดด้วย

ในการลดคำกำกวมในการแปลภาษาไทยเป็นภาษาไทยล้านนา ซึ่งจะลดความกำกวมของโครงสร้างประโยค และลำดับของคำศัพท์โดยใช้แบบจำลองฮิดเด้นมาร์คอฟ และวิเทอร์บีอัลกอริทึม(Viterbi Algorithm) ซึ่งวิเทอร์บีอัลกอริทึมจะมี ขั้นตอนดังภาพที่ 3.3

```
For i=1 to N do
    SEQSCORE(i,1)= PROB(w_1 | L_i) * PROB(L_i | ø)
    BACKPTR(i,1)=0

For t=2 to T
    For i=1 to N
        SEQSCORE(i,t)= Max_{j=1,N} (SEQSCORE(j,t-1)
                            * PROB(L_i | L_j)) * PROB(w_t | L_i)
        BACKPTR(i,t)=index of j that gave the max above

C(T)=i that maximizes SEQSCORE(i,T)
For i=T-1 to 1 do
    C(i)=BACKPTR(C(i+1),i+1)
```

ภาพที่ 3.3 : วิเทอร์บีอัลกอริทึม(Viterbi Algorithm) สำหรับการลดคำกำกวมในการแปลภาษาไทยเป็นภาษาไทยล้านนา [11]

จากอัลกอริทึมมีตัวแปรที่เกี่ยวข้องดังนี้

- -T คือ จำนวนคำในโครงสร้างประโยคภาษาไทยล้านนาที่มีการระบุชนิดของคำ ที่ต้องการเลือกโครงสร้าง
- - N คือ จำนวนชนิดของคำภาษาไทยล้านนาที่พบในประโยคที่ต้องการแปล
- - SEQSCORE คือ อะเรย์ขนาด N*T ที่เก็บค่าความน่าจะเป็นชุดหนึ่งที่มีการเปลี่ยนสถานะ(คำ) ตั้งแต่สถานะเริ่มต้นไปจนถึงสถานะ(คำ) ปัจจุบัน
- - BACKPTR คือ อะเรย์ขนาด N*T ที่เก็บเส้นทางที่มีค่าความน่าจะเป็นสะสมสูงสุด
- - w คือ คำในโครงสร้างประโยคภาษาไทยล้านนา มีการระบุชนิดของคำไทยล้านนาที่ต้องการแปล
- - L คือ ชนิดของคำภาษาไทยล้านนาในประโยคที่ต้องการแปล
- - PROB(w_i|L_j) คือ ความน่าจะเป็นของคำภาษาไทยล้านนา w_i ที่มีชนิดของคำภาษาไทยล้านนาเป็นเป็น L_j ซึ่งคำนวณจากจากสูตร

$$\mathrm{PROB}(w_i|L_j) \cong \frac{\text{Count(w at position i and L at position j)}}{\text{Count(L at position j)}}$$

- $\mathrm{PROB}(L_i|L_j)$ คือ ความน่าจะเป็นที่เมื่อเกิดชนิดของคำภาษาไทยล้านนา L_j แล้วจะเกิดชนิดของคำภาษาไทยล้านนา L_i ตามหลังซึ่งคำนวณจากสูตร

$$\mathrm{PROB}(L_i|L_j) \cong \frac{\text{Count(L at position i and L at position j)}}{\text{Count(L at position j)}}$$

- C คือ ลิสต์ที่มีขนาดเท่ากับจำนวนคำในประโยคภาษาไทยล้านนาที่ต้องการเลือกโครงสร้าง ซึ่งจะเก็บลำดับของคำภาษาไทยล้านนาที่มีค่าความน่าจะเป็นสูงสุด

คลังข้อมูลที่นำมาคำนวณค่าสถิติ จะเตรียมคลังข้อมูลภาษาไทยล้านนาที่มีการกำกับชนิดของคำ ในประโยคไทยล้านนา วิเทอร์บีอัลกอริทึมจะคำนวณเพื่อหาโครงสร้างของประโยคล้านนาทั้งหมดทุกโครงสร้าง จากขั้นตอนของการปรับโครงสร้างล้านนาในไวยากรณ์ปริวรรตก่อนหน้า ผลลัพธ์ของขั้นตอนนี้จะได้ลำดับของคำในแต่ละโครงสร้างประโยคล้านนา และจะนำค่าสถิติที่คำนวณได้ มาเลือกโครงสร้างที่มีความน่าจะเป็นสูงสุด จะทำให้ได้โครงสร้างที่เป็นผลลัพธ์ของการแปลภาษาไทยเป็นภาษาไทยล้านนา

3.7 การทดลองและการวัดประเมินผลการทดลอง

เนื่องจากภาษาไทยและภาษาไทยล้านนามีความใกล้เคียงกันในลักษณะโครงสร้างของภาษาและคำที่ใช้งานในประโยค เพราะภาษาไทยล้านนาเป็นภาษาถิ่น ดังนั้นในการออกแบบการทดลองเพื่อวัดประสิทธิภาพของการแปลจึงมีการออกแบบการทดลอง วิธีการเพื่อตรวจสอบผลลัพธ์ของการแปล ได้แก่

การแปลโดยใช้ไวยากรณ์ปริวรรต วิเคราะห์และจัดรูปแบบโครงสร้างของประโยคล้านนาก่อนเข้าสู่กระบวนการเลือกโครงสร้างและลดความกำกวมของคำศัพท์โดย วิเทอร์บีอัลกอริทึม

การวัดประเมินผลการทดลอง ผลจากการทดลองจะนำมาวัดประสิทธิภาพการแปล โดยการคำนวณค่าร้อยละ จากประโยคที่แปลถูกต้องต่อประโยคที่นำมาทดสอบทั้งหมด ดังสมการ

$$\text{ประสิทธิภาพการแปล (\%)} = \frac{X}{N} \times 100$$

เมื่อกำหนดให้

ตัวแปร X คือ จำนวนของประโยคที่แปลถูกต้อง

ตัวแปร N คือ จำนวนของประโยคที่นำมาทดสอบทั้งหมด

บทที่ 4
การประเมินผลความถูกต้องของการแปล

จุดประสงค์ของงานวิจัยนี้คือการแปลภาษาไทยเป็นภาษาไทยล้านนา โดยใช้วิธีการเลือกความหมายจากพจนานุกรมไทยล้านนา วิเคราะห์โครงสร้างประโยคภาษาไทยด้วยไวยากรณ์ไม่พึ่งบริบท ประยุกต์เทคนิคทางสถิติ (Probabilistic Context free grammar : PCFG) ปรับโครงสร้างไวยากรณ์ให้เป็นโครงสร้างไวยากรณ์ไทยล้านนาด้วย ไวยากรณ์ปริวรรต (Transformation Grammar) และลดความกำกวมด้วยวิธีการของแบบจำลองฮิดเด้นมาร์คอฟและวิเทอร์บีอัลกอริทึม (Viterbi algorithm)

ข้อมูลที่นำมาทดสอบเป็นข้อมูลที่รวบรวมจากข้อมูลที่มีการบันทึกไว้ในเอกสาร โดยผู้เชี่ยวชาญภาษาไทยล้านนา ซึ่งได้นำมารวบรวม ซึ่งการที่นำข้อมูลชุดนี้มาทำการทดสอบเนื่องจากเป็นข้อมูลที่น่าสนใจและสามารถตรวจสอบผลลัพธ์ความถูกต้องของประโยคภาษาไทยล้านนาที่เป็นผลลัพท์

ตัวอย่างข้อมูลที่นำมาทำการทดลองประกอบไปด้วย นิทานพื้นบ้าน 3 เรื่อง จำนวน 121 ประโยค คำประโยคทดสอบที่ถูกต้องตามไวยากรณ์ภาษาไทยจำนวน 85 ประโยค

4.1 ผลการวิจัย

การทดสอบโดยการนำประโยคทดสอบเข้าสู่กระบวนการตัดคำโดยใช้เครื่องมือตัดคำ (KU-CUT) ซึ่งเป็นเครื่องมือที่ตัดคำตัดคำด้วยเทคนิคการเรียนรู้จากข้อมูลแบบไม่ใช้ตัวอย่างร่วมกับการใช้พจนานุกรมและคำ จากนั้นทำการแปลโดยค้นหาคำแปลภาษาไทยล้านนาจากพจนานุกรม ไทย-ล้านนา หากในกรณีที่ไม่พบคำแปลในพจนานุกรมจะใช้ วิธีการปริวรรตคำไทยล้านนาตามกฎของโครงสร้างของการเขียนภาษาไทยล้านนา จากนั้นนำเข้าสู่ไวยากรณ์ปริวรรตปรับให้เป็นโครงสร้างประโยคไทยล้านนา จากนั้นจะลดความกำกวมของโครงสร้างประโยคและลำดับของคำล้านนา โดยวิธีการวิเทอร์บีอัลกอริทึม ผลลัพท์ก็จะได้ประโยคไทยล้านนาที่โครงสร้างและลำดับคำที่ถูกต้อง

การวัดประสิทธิภาพของการแปล โดยการคำนวณค่าร้อยละ จากประโยคที่แปลถูกต้องต่อประโยคที่นำมาทดสอบทั้งหมด ดังสมการ

$$\text{ประสิทธิภาพการแปล (\%)} = \frac{X}{N} \times 100$$

เมื่อกำหนดให้

ตัวแปร X คือ จำนวนของประโยคที่แปลถูกต้อง

ตัวแปร N คือ จำนวนของประโยคที่นำมาทดสอบทั้งหมด

ผลลัพท์จากการวัดประสิทธิภาพของการแปลภาษาไทยเป็นภาษาไทยล้านนาของงานวิจัยนี้ ได้แสดงผลลัพท์ดังตารางที่ 4.1

ชื่อเอกสาร	จำนวนประโยค	แปลถูกต้อง	อัตราส่วนร้อยละ
นิทานพื้นบ้าน เรื่อง ปูกับนกยาง	29	21	72.41
นิทานพื้นบ้าน เรื่อง คนไม่รู้บุญคุณ	27	20	74.07
นิทานพื้นบ้าน เรื่อง กินข้าวอร่อยทุกมื้อ	65	51	78.46
ประโยคทดสอบ	85	70	87.5
รวมทั้งสิ้น	206	162	78.64

ตารางที่ 4.1 : ประสิทธิภาพของการแปลภาษาไทยเป็นภาษาไทยล้านนา

4.2 การอภิปรายผล

จากการแปลภาษาไทยเป็นภาษาไทยล้านนาด้วยเลือกความหมายจากพจนานุกรมไทยล้านนา วิเคราะห์โครงสร้างประโยคภาษาไทยด้วยไวยากรณ์ไม่พึ่งบริบท ประยุกต์เทคนิคทางสถิติ (Probabilistic Context free grammar : PCFG) ปรับโครงสร้างไวยากรณ์ให้เป็นโครงสร้างไวยากรณ์ไทยล้านนาด้วย ไวยากรณ์ปริวรรต (Transformation Grammar) และลดความกำกวมด้วยวิธีการของแบบจำลองฮิดเด้นมาร์คอฟและวิเทอร์บีอัลกอริทึม (Viterbi algorithm) ผลลัพธ์จะได้ประโยคภาษาล้านนาซึ่งเป็นภาษาเป้าหมาย จากการทดลองพบว่า จากประโยคทั้งสิ้น 206 ประโยคที่นำมาทดลองมีประสิทธภาพในการแปล 78.64%

จากการทดลองพบว่า การแปลภาษาไทยเป็นภาษาไทยล้านนาด้วยวิธีเลือกคำศัพท์จากพจนานุกรมนั้น คำไทย 1 คำมักจะแปลเป็นคำไทยล้านนาได้หลายคำ ทำให้ส่งผลต่อการที่จะนำคำศัพท์มาสร้างประโยคที่เป็นไปได้ทั้งหมด เมื่อเข้าสู่ขั้นตอนการลดความกำกวมด้วยวิเทอร์บีอัลกอริทึมก็จะใช้เวลาการประมวลผลค่อนข้างนานเพราะจะต้องคำนวนหาค่าทางสถิติทุกประโยคเพื่อเรียงประโยคผลลัพท์ในการแปลด้วย

อย่างไรก็ตามในการปรับโครงสร้างไวยากรณ์ไทยให้เป็นโครงสร้างไวยากรณ์ไทยล้านนาด้วยไวยากรณ์ปริวรรตเพิ่มพูน(Thai-Thai lanna transformation rule) นั้นจะทำให้ประโยคบางประโยคมีความสมบูรณ์มากยิ่งขึ้น ตัวอย่างเช่น

ตัวอย่างผลลัพท์ของการแปลที่ผ่านไวยากรณ์ปริวรรต

วลี "อร่อยมาก" จะประกอบด้วยคำว่า "อร่อย" ที่มีคำแปลภาษาไทยล้านนาคือคำว่า ลำ(ᩃᩣ᩠ᨾ) และคำว่า "มาก" มีคำแปลภาษาไทยล้านนา หลายคำแปล คือ ขนาด(ᨡᨶᩣ᩠ᨯ) , นัก(ᨶᩢ᩠ᨠ) , ชาด(ᨩᩣ᩠ᨯ) ซึ่งเมื่อแปลออกมาแล้วจะได้ผลลัพทคือ

ลำขนาด(ᩃᩣ᩠ᨾ ᨡᨶᩣ᩠ᨯ)

ลำนัก(ᩃᩣ ᨶᩢ᩠ᨠ)

ลำชาด(ᩃᩣ ᨩᩣ᩠ᨯ)

ซึ่งเมื่อปรับโครงสร้างด้วย ไวยากรณ์ปริวรรต ที่ตรงกับกฎของไวยากรณ์ปริวรรตแล้ว จะทำให้คำว่า ลำชาด(ᩃᩣ ᨩᩣ᩠ᨯ) มีการสลับตำแหน่งของคำเป็น ชาดลำ(ᨩᩣ᩠ᨯ ᩃᩣ) ซึ่งเป็นผลลัพท์คำแปลที่ถูกต้องนั่นเอง

สาเหตุของความผิดพลาดของการแปลสามารถจำแนกสาเหตุได้ดังนี้

1.ในการตัดคำและกำกับชนิดของคำ มีผลทำให้การแปลผิดพลาดเช่น ประโยค “ป้าแก้วทำกับข้าวเลี้ยงแขก” เมื่อคำว่า “แก้ว” คือคำนาม เมื่อค้นหาในพจนานุกรมแล้วได้ คำแปลของคำว่าแก้วคือ รัตน จึงได้ประโยคนี้เป็น “ᨸ᩶ᩣ ᩁᨲ᩠ᨶ ᨶᩮᩁᨾᩥ᩠ᨲ ᨣᩢ᩠ᨷᨡᩮᩢ᩶ᩣ ᩃ᩠ᨿ᩶ᨦ ᨡᩯ᩠ᨠ” อ่านว่า ป้ารัตนะเนรมิตรกับข้าวเลี้ยงแขก

2. ในขั้นตอนการแปลงรูปเป็นคำไทยล้านนาหากไม่พบในพจนานุกรม มีความผิดพลาดเนื่องจากคำไทยคำนั้นมีโครงสร้างที่ซับซ้อนเช่นคำว่า “เนิบนาบ” จะพบว่าคำว่าเนิบมีตัวสะกดคือ “บ” แต่คำถัดไปคือ “น” จากคำว่านาบ ซึ่งทำให้เกิดความผิดพลาดในการแปลงรูปเพราะเจอกฎของการแปลงคำว่า “บน”

3. ในขั้นตอนของการคำนวณด้วยวิเทอบีอัลกอริทึม หากคำศัทพ์และชนิดของคำดังกล่าวไม่มีใน คลังข้อมูลของประโยคภาษาไทยล้านนาที่ใช้ในการทดสอบ จะทำให้ส่งผลต่อผลลัพท์ของการแปลที่ควรจะเป็น และทำให้การแปลความหมายมีความคลาดเคลื่อนกับประโยคที่ควรจะเป็นได้

4.3 อุปสรรคในการทำงาน

อุปสรรคบางประการส่งผลให้การแปลผิดพลาด อุปสรรคดังกล่าวได้แก่

1. ความไม่ครอบคลุมของคำศัพท์ในพจนานุกรม การค้นหาคำศัพท์ในพจนานุกรม คำไทยล้านนาบางคำศัพท์เป็นคำศัพท์เฉพาะ ทำให้เมื่อนำไปแทนที่คำไทยมีความหมายที่ผิดเพี้ยนไป คำไทยบางคำก็มีคำศัพท์ไทยล้านนาเพียง 1 คำที่เป็นคำเฉพาะ แต่เมื่อแทนความหมายของคำศัพท์นั้นในประโยค ประโยคดังกล่าวก็มีความผิดเพี้ยนในความหมาย

2. คำไทยบางคำแปลเป็นคำไทยล้านนาได้มากเกินไป คำศัพท์ไทยล้านนาที่มีมากเกินไปทำให้ส่งผลต่อการคำนวณที่จะใช้เวลาในการแปลประโยคนั้นนาน เช่น ประโยค “พ่อกินข้าว” พบว่า

คำว่า พ่อ มีคำศัพท์ 1 คำในพจนานุกรมคือ พ่อ (ᨻᩴ᩵)

คำว่า กิน มีคำศัพท์ 7 คำในพจนานุกรมคือ กิน (ᨠᩥ᩠ᨶ), ภุญชะ (ᨽᩩᨬ᩠ᨩ), ย้ำ (ᨿᩣᩴ᩶), รูดเน่า (ᩁᩪᨯᨶᩮᩢ᩵ᩣ), เสพ (ᩈᩮ᩠ᨷ), เสวย (ᩈᩮ᩠ᩅ᩠ᨿ), ฉัน (ᨨᩢ᩠ᨶ)

คำว่า คำว่า ข้าว มีคำศัพท์ 2 คำในพจนานุกรมคือ ภัตตะ (ᨽᨲ᩠ᨲ), เข้า (ᨡᩮᩢ᩶ᩣ)

เมื่อนำมาสร้างประโยค ก็จะมีประโยคทั้งหมด 14 ประโยคได้แก่

ᨸᩪ᩵ ᨠᩥ᩠ᨶ ᨣᩤ᩠ᨲ

ᨸᩪ᩵ ᨣᩣ᩠ᨬ= ᨣᩤ᩠ᨲ

ᨸᩪ᩵ ᨿᩣ᩶ ᨣᩤ᩠ᨲ

ᨸᩪ᩵ ᨡᩪ᩠ᨪᩮ᩠ᨦ᩶ ᨣᩤ᩠ᨲ

ᨸᩪ᩵ ᨩᩮᩢ᩠ᨿ ᨣᩤ᩠ᨲ

ᨸᩪ᩵ ᨩᩮᩥᩬ᩠ᨿ ᨣᩤ᩠ᨲ

ᨸᩪ᩵ ᨡ᩠ᨿ᩶ ᨣᩤ᩠ᨲ

ᨸᩪ᩵ ᨠᩥ᩠ᨶ ᨡᩮᩢ᩶ᩣ

ᨸᩪ᩵ ᨣᩣ᩠ᨬ= ᨡᩮᩢ᩶ᩣ

ᨸᩪ᩵ ᨿᩣ᩶ ᨡᩮᩢ᩶ᩣ

ᨸᩪ᩵ ᨡᩪ᩠ᨪᩮ᩠ᨦ᩶ ᨡᩮᩢ᩶ᩣ

ᨸᩪ᩵ ᨩᩮᩢ᩠ᨿ ᨡᩮᩢ᩶ᩣ

ᨸᩪ᩵ ᨩᩮᩥᩬ᩠ᨿ ᨡᩮᩢ᩶ᩣ

ᨸᩪ᩵ ᨡ᩠ᨿ᩶ ᨡᩮᩢ᩶ᩣ

ดังนั้นจำนวนประโยค ที่คำนวณทั้งหมดจะเป็นผลคูณของ คำไทยล้านนาที่พบในแต่ละคำไทย จึงทำให้การคำนวณต้องทำทุกประโยคส่งผลให้ใช้เวลานาน

3. ความขาดแคลนคลังข้อมูลภาษา(corpus) ไทยล้านนาสร้างได้ยาก การลดความกำกวมด้วยวิธีการของวิเทอร์บีอัลกอริทึม จำเป็นต้องพึ่งพา คลังข้อมูลภาษา ซึ่งภาษาไทยล้านนาเป็นภาษาที่มีการศึกษาและเชี่ยวชาญกันเฉพาะกลุ่ม จึงทำให้ขั้นตอนในการสร้างคลังข้อมูลภาษาไทยล้านนาเป็นไปด้วยความลำบาก ข้อจำกัดดังกล่าวทำให้ประโยคที่ต้องการข้อมูลจากคลังข้อมูลภาษาไทยล้านนา ในการลดความกำกวมนั้น ไม่ครอบคลุมในประโยคบางประโยคได้ทำให้ส่งผลต่อผลลัพธ์ของการแปลภาษาไทยล้านนา

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

วิทยานิพนธ์ฉบับนี้ได้นำเสนอการแปลภาษาไทยเป็นภาษาไทยล้านนา โดยใช้ความหมายจากพจนานุกรมไทยล้านนา วิเคราะห์โครงสร้างประโยคภาษาไทยด้วยไวยากรณ์ไม่พึ่งบริบท ประยุกต์เทคนิคทางสถิติ (PCFG) ปรับโครงสร้างไวยากรณ์ให้เป็นโครงสร้างไวยากรณ์ไทยล้านนาด้วยไวยากรณ์ปริวรรต (Transformation Grammar) และลดความกำกวมด้วยวิธีการของแบบจำลองฮิดเด้นมาร์คอฟและวิเทอร์บีอัลกอริทึม (Viterbi algorithm)

ภาษาไทยและภาษาไทยล้านนามีโครงสร้างไวยากรณ์ที่ใกล้เคียงกัน แต่ก็มีบางโครงสร้างที่มีความแตกต่าง ไวยากรณ์ปริวรรต (Transformation Grammar) จะช่วยทำให้การแปลภาษาไทยเป็นภาษาไทยล้านนามีความถูกต้องและใกล้เคียงกับคำแปลภาษาไทยล้านนามากยิ่งขึ้น

ข้อมูลที่ใช้ในการวัดประสิทธิภาพของการแปลภาษาไทยเป็นภาษาไทยล้านนาประกอบด้วยนิทานจำนวน 3 เรื่อง 121 ประโยค, ประโยคทดสอบจำนวน 85 ประโยค พบว่ามีประสิทธิภาพของการแปล 78.64%

โดยมีสาเหตุที่ทำให้การแปลมีความผิดพลาดคือ การตัดคำและกำกับชนิดของคำมีผลทำให้การแปลผิดพลาด , ขั้นตอนการแปลงรูปเป็นคำไทยล้านนา มีความผิดพลาดเนื่องจากคำไทยคำนั้นมีโครงสร้างที่ซับซ้อน , การคำนวณด้วยวิเทอบีอัลกอริทึม หากคำศัทพ์และชนิดของคำไม่มีในคลังข้อมูลของประโยคภาษาไทยล้านนา จะทำให้ส่งผลต่อผลลัพท์ของการแปล

5.2 ข้อเสนอแนะ

จากปัญหาที่พบในงานวิจัย เนื่องจากภาษาเป้าหมายนั้นคือภาษาไทยล้านนา เป็นภาษาถิ่นเหนือซึ่งแม้ว่าจะมีการเผยแพร่องค์ความรู้ภาษาไทยล้านนามากยิ่งขึ้นในปัจจุบัน แต่ก็ยังถือว่าเป็นวงแคบและยังมีผู้ที่เชี่ยวชาญค่อนข้างจำกัด ส่งผลถึงเอกสาร และข้อมูลที่เกี่ยวข้องกับ ภาษาไทยล้านนาที่ออกสู่สาธารณะค่อนข้างที่จะมีน้อย ทำให้ในบางกระบวนการที่ต้องการข้อมูลเหล่านั้นเช่น การสร้างคลังข้อมูลภาษาทำได้ค่อนข้างยากดังนั้นควรส่งเสริมให้มีการเรียนรู้ภาษาไทยล้านนาในวงกว้างเพื่อจะได้มีข้อมูล เอกสารต่างๆ ออกสู่สาธารณะต่อไป

ส่วนของโปรแกรมในการสร้างประโยคที่เป็นไปได้ทั้งหมด หากพบว่าคำศัพท์ที่ได้จากการค้นหาในพจนานุกรมปริมาณมากจะทำให้ใช้เวลาในการประมวลผลค่อนข้างนาน เนื่องจากต้องประมวลผลคำนวณข้อมูลทางสถิติของทุกประโยคตามไปด้วย

ด้วยทรัพยาการคลังภาษาไทยและภาษาไทยล้านนา ค่อนข้างมีจำกัดและไม่ได้เจาะจงเฉพาะขอบเขตเฉพาะด้าน ทำให้เกิดปัญหาในข้อมูลเบาบาง ทำให้เมื่อนำมาใช้ในการวิเคราะห์โครงสร้างประโยคภาษาไทยด้วยไวยากรณ์ไม่พึ่งบริบท ประยุกต์เทคนิคทางสถิติ และ การลดความกำกวมด้วยวิธีการของแบบจำลองฮิดเด้นมาร์คอฟและวิเทอร์บีอัลกอริทึม ไม่ครอบคลุมดังเช่นสภาพแวดล้อมของการใช้งานจริง ดังนั้นหากทรัพยากรคลังข้อมูลทั้งภาษาไทยและภาษาไทยล้านนามีการรวบรวมที่เฉพาะเจาะจง และมีปริมาณที่ครอบคลุมทุกบริบท คาดว่าจะทำให้ผลลัพธ์ของการแปลภาษาไทยเป็นภาษาไทยล้านนามีความถูกต้องมากยิ่งขึ้น

บรรณานุกรม

[1] ศรีเนียน สวัสดิ์. **ตำราเรียนหนังสือ ภาษาไทยล้านนา** เชียงใหม่ : ร้านประเทืองวิทยา, 2540

[2] พระมหามิลินท์ อนาคาริโก. **คู่มือเรียนภาษาไทยล้านนา 24 ชั่วโมงด้วยตนเอง.** ค้นเมื่อ 24 กุมภาพันธ์ 2554 , จาก http://www.bmwit.net/word-pdf/Test-003.pdf

[3] อัศนีย์ ก่อตระกูล. **การประมวลภาษามนุษย์ด้วยคอมพิวเตอร์**. กรุงเทพฯ: หน่วยปฏิบัติการวิจัยเชี่ยวชาญเฉพาะการประมวลผลภาษยธรรมชาติและเทคโนโลยีสารสนเทศอัจฉริยะ ภาควิชาวิศวะกรรมศาสตร์ คณะวิศวะกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์, 2549.

[4] พุธยดี ศิริแสงตระกูล และ นิกร ยาพรม, "การแปลภาษาล้านนาเป็นภาษาไทยกลาง", วารสารวิจัย มหาวิทยาลัยขอนแก่น, มีนาคม 2552 หน้า 264-274.

[5] โสภัณ พรมโสดา และ พุธยดี ศิริแสงตระกูล, "การแปลงอักษรธรรมอีสานเป็นภาษาไทย", The 7th National Conference on Computing and Information Technology (NCCIT2011),พฤษภาคม 2554 หน้า 503-509.

[6] Ola Mohammad Ali, Mahmoud GadAlla, and Mohammad Said Abdelwahab, "Improving Machine Translation using Hybrid Dictionary-Graph Based Word Sense Disambiguation with Semantic and Statistical Methods" *International Journal of Computer and Electrical Engineering,* Vol.1, No.5, 2009, pp.1793 –8163.

[7] เรืองเดช ปันเขื่อนขัติย์. **การวิเคราะห์ประโยคภาษาไทย**. สารภาษาไทยและวัฒนธรรมไทย ภาควิชาภาษาไทย คณะมนุษยศาสตร์ มหาวิทยาลัยเกษตรศาสตร์, 2550

[8] ทรงศักดิ์ ปรางค์วัฒนากุล, บรรณาธิการ. **ข้อตกลงในการปริวรรตอักษรธรรมล้านนาเป็นอักษรไทยกลาง**. เชียงใหม่: โรงพิมพ์ช้างเผือก; 2529.

[9] ยืน ภู่วรรณ และ ชัยยงค์ วงศ์ชัยสุวัฒน์. **การประมวลผลภาษาธรรมชาติ(Natural Language Processing)**. กรุงเทพฯ: ศูนย์เทคโนโลยีอิเล็กทรอนิกส์ และคอมพิวเตอร์แห่งชาติ กระทรวงวิทยาศาสตร์และเทคโนโลยีและสิ่งแวดล้อม; 2535. หน้า 76-84

[10] ยศวัธน์ แสนสิงห์ และ อัครา ประโยชน์, "การรู้จำชื่อเฉพาะภาษาไทยโดยใช้แบบจำลองฮิดเดนมาร์คอฟ", *Proceedings of the Second Conference on Knowledge and Smart Technologies* 2010 *(KST-2010)*, Chonburi Thailand, 24-25 July 2010.

บรรณานุกรม(ต่อ)

[11] Allen James, Natural **language understamding**,2nd ed,1994 , pp.202

ภาคผนวก

ภาคผนวก ก
ตัวอย่างผลลัพท์ของการทดสอบการแปล

1.นิทานพื้นบ้านเรื่อง ปูกับนกยาง

ปูกับนกยางนี้มาอยู่หนองหนึ่ง หนองนั้นกลางฤดูฝนก็มีน้ำ แต่หน้าแล้งนี้น้ำไม่ค่อยมี หากว่าน้ำแห้งก็จะตายเลิกพอดี นกยางตัวหนึ่งบินมาบอกว่า มาอยู่นี้ก็จะตาย น้ำแห้งต่อจากนั้น นกยางก็รับอาสาย้ายที่อยู่อีกหนองหนึ่ง ปูและปลาก็จะมาตกลงกันว่าใครจะไปก่อน ปลาก็บอกว่าข้าจะขอไปก่อน นกยางก็คาบพาปลาไป บินไปปล่อยลำบากไปอย่างไรไปทั้งวันพอดี ก็กลับมาเหลือแต่ปู นกยางปากแหลมคีบปูไม่แน่น คีบหลุด คีบหลุด ไม่รู้จะทำอย่างไร ก็คีบจนกระทั่งหลังเป็นรอยบุบ ปูก็ว่าข้าพเจ้าขอเอาก้ามคีบคอนกยางเถอะ นกยางก็ตกลง พอไปถึงที่เอาปลาไว้ ปูก็เห็นปลา ที่นกยางกินเหลือแต่ก้าง ปูก็ตกใจกลัว ว่าจะถูกกินอย่างปลา ก็เลยเอาก้ามคีบที่คอ นกยางถึงกับร้อง ออกออก นกยางเลยมีนิสัยร้อง ออกออก จนถึงเดี๋ยวนี้

ผลลัพท์การแปล นิทานพื้นบ้านเรื่อง ปูกับนกยาง

[illegible]

2.นิทานพื้นบ้านเรื่อง คนไม่รู้บุญคุณ

มีหนองน้ำอยู่หนองหนึ่ง น้ำในหนองนั้นก็แห้ง ก็มีแต่เต่าอยู่ในหนองน้ำนั้นตัวหนึ่ง พระยาหงส์มาเห็นใส่ก็เลยว่า เต่าหนีจากหนองน้ำนี้ไหม ข้าพเจ้าไปไม่ได้ ไม่มีปีก ข้าจะช่วยกันหามไป ถ้าว่าเต่าจะอยู่เฝ้าหนองน้ำนี้ ไม่มีอะไร ไม่มีทางที่น้ำจะมี เต่าก็บอกข้าจะไปยังไง ไม่ยาก หาไม้ท่อนมา หงส์สองตัวก็ไปหาไม้ท่อนมา แล้วให้เต่านั้นคาบไม้อยู่กึ่งกลาง แล้วช่วยกันหามสองข้าง แล้วพากันบินไป ไปผ่านบ้านผ่านเมือง พอเด็กเห็นก็ร้อง หงส์หามเต่า ทีนี้เต่าไม่ยอม มันก็เลยว่า มันจะใช่อย่างไร เต่าหาบหงส์ต่างหาก เต่ามันก็เลยร่วงไป นี่แหละเขาว่า ปากไม่ดี ขาช่วยแล้ว ยังไม่รู้บุญคุณเขา

ผลลัพท์การแปล นิทานพื้นบ้านเรื่อง คนไม่รู้บุญคุณ

[illegible]

ภาคผนวก ข
บทความวิจัยและผลงานที่ได้ตีพิมพ์

NCCIT 2011

THE 7TH NATIONAL CONFERENCE ON COMPUTING AND INFORMATION TECHNOLOGY

PROCEEDINGS OF NCCIT 2011
THE 7TH NATIONAL CONFERENCE ON COMPUTING AND INFORMATION TECHNOLOGY
11-12 MAY 2011
WWW.NCCIT.NET

FACULTY OF INFORMATION TECHNOLOGY
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

VOLUME 2

IEEE THAILAND SECTION

บทความวิจัย

การประชุมทางวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ครั้งที่ 7

11-12 พฤษภาคม 2554

คณะเทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

VOLUME 2

THE 7TH NATIONAL CONFERENCE ON COMPUTING AND INFORMATION TECHNOLOGY

PROCEEDINGS OF NCCIT 2011

THE 7TH NATIONAL CONFERENCE ON COMPUTING AND INFORMATION TECHNOLOGY

11-12 MAY 2011

FACULTY OF INFORMATION TECHNOLOGY
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK (KMUTNB)
BANGKOK, THAILAND

WWW. NCCIT.NET

IEEE
THAILAND SECTION

บทความวิจัย

การประชุมทางวิชาการระดับชาติด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
ครั้งที่ 7

11-12 พฤษภาคม 2554

KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK

คณะเทคโนโลยีสารสนเทศ

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

Room 7		
Session 6 : Image Processing and Pattern Recognition		
Time/Paper-ID	**Title/Author**	**Page**
13.00-13.20 **NCCIT2011-158**	**Comparison of vehicle license plate localization algorithm: EP & WEP** *Suchitra Audulkasem, Jitdumrong Preechasuk, Piticha Paewchompoo, and Saowaluk Jirakulkanok*	784
13.20-13.40 **NCCIT2011-194**	**Automatic Detection and Counting Scalp Hairs** *Sumaree Arayasombat and Nongluk Covavisaruch*	790
13.40-14.00 **NCCIT2011-187**	**Measuring of a Vitiligo Lesion Area on a Curved Surface Using Digital Image Processing** *Bharima Clangphukhieo and Nongluk Covavisaruch*	795
Session 7: Web Application, Web Service, Information Retrieval, Natural Language Processing, and Ontology		
Time/Paper-ID	**Title/Author**	**Page**
14.00-14.20 **NCCIT2011-191**	**Developing an Ontology Knowledge Based for Automatic Online News Analysis** *Wichuda Chotirat, Pudsadee Boonrawd, and Sageemas Na Wichian*	800
14.20-14.40 **NCCIT2011-33**	**A Development of Ontology Prototyping for Thai Massage Therapy Searching** *Wiraiwan Sanchana, Orasa Tetiwat, Nattavadee Hongboonmee, and Marut Buranarach*	806
14.40-15.00	*Coffee Break*	
15.00-15.20 **NCCIT2011-161**	**Reducing Ambiguity in Thai Text to Thai Lanna Text Translation System using Viterbi Algorithm** *Prathan Comejina and Ponrudee Netisopakul*	812
15.20-15.40 **NCCIT2011-18**	**A Web-based Single Sign-on (SSO) using SAML 2.0** *Tatchai Russameroj, Pornchai Mongkolnam, and Kriengkrai Porkaew*	818
15.40-16.00 **NCCIT2011-24**	**The System Judging Criteria of the Web Design Competition Case Study Development of Skill Department** *J.Jamsai, U.Rodrueng, and Somboon Supattarakulchai*	824
16.00-16.20 **NCCIT2011-40**	**(Server Management System On CentOS Linux Server Case Study: Website Skill Competition (Web Design),Development of Skill Department)** *Nut Pornsopon, Arisa Getgosol, and Anumas Sangsawang*	830
16.20-16.40 **NCCIT2011-182**	**Web Service System for Social Network, Facebook** *Prapatsorn Sripadet and Cholatip Yawut*	836

การลดความกำกวม ในการแปลภาษาไทยเป็นภาษาไทยล้านนาด้วยวิเทอร์บีอัลกอริทึม
Reducing Ambiguity in Thai Text to Thai Lanna Text Translation System using Viterbi Algorithm

ประธาน คำจินะ (Prathan Comejina) [1] และ *พรฤดี เนติโสภากุล (Ponrudee Netisopakul)* [2]
[1,2]*คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง*
pcomejina@gmail.com, ponrudee@it.kmitl.ac.th

บทคัดย่อ

การแปลภาษาไทยเป็นภาษาไทยล้านนา จะพบปัญหาการจับคู่ระหว่างคำไทยหนึ่งคำ กับ คำไทยล้านนาได้หลายคำ ดังนั้น งานวิจัยนี้จึงนำเสนอ วิธีการลดความกำกวมในการเลือกคำไทยล้านนา เพื่อการแปลภาษาไทยเป็นภาษาไทยล้านนา ด้วยวิธีการของแบบจำลองฮิดเดนมาร์คอฟและวิเทอร์บีอัลกอริทึม ร่วมกับคลังคู่ประโยคไทย-ไทยล้านนา จากการทดลองเบื้องต้นโดยใช้วิเทอร์บีอัลกอริทึม พบว่ามีความถูกต้องของการแปลเพิ่มขึ้นร้อยละ 60

คำสำคัญ: แบบจำลองฮิดเดนมาร์คอฟ วิเทอร์บีอัลกอริทึม การแปลภาษาด้วยคอมพิวเตอร์

To translate Thai language to Thai Lanna language, the result shows one Thai word can be translated into a few Thai Lanna words. To select the correct translation, this paper proposes to use hidden markov model and viterbi algorithm with Thai – Thai Lanna corpus. The accuracy of reducing ambiguity in translation using viterbi algorithm increases 60 percent.

Keyword: Hidden Markov Model, Viterbi Algorithm, Machine Translation

1. บทนำ

ภาษาไทยล้านนา เป็นภาษาถิ่นของคนไทยในภาคเหนือตอนบน พัฒนามาจากอักษรมอญ เมื่อประมาณ พ.ศ.1919 ซึ่งเป็นช่วงเดียวกันกับการเกิดอักษรภาษาไทย ภาษาไทยล้านนาจึงเป็นภาษาเก่าแก่ที่มีคุณค่าทางวัฒนธรรม และเป็นเอกลักษณ์เฉพาะตัวของคนไทยในภาคเหนือ ปัจจุบันผู้เชี่ยวชาญการใช้ภาษาไทยล้านนามีน้อยลง จึงควรที่จะอนุรักษ์และสืบทอดภาษาไทยล้านนาไม่ให้สูญหาย

งานด้านการแปลภาษาด้วยคอมพิวเตอร์ โดยทั่วไปจะพบปัญหาความกำกวม ในระดับคำและระดับโครงสร้างประโยค เช่น งานวิจัยการแปลล้านนาเป็นภาษาไทย [1] เป็นงานวิจัยที่มีความคล้ายคลึงกับงานวิจัยของเรา ตรงการใช้คู่ภาษาสำหรับการแปลเดียวกัน งานวิจัย [1] พบความกำกวมในระดับโครงสร้างประโยค และใช้วิธีข่ายงานเพิ่มขยายช่วยเลือกโครงสร้างประโยคที่ถูกต้อง ส่วนงานวิจัย [2] ก็พบปัญหาการแปลในระดับโครงสร้างและแก้ปัญหานี้ โดยใช้วิเทอร์บีอัลกอริทึมช่วยเลือกโครงสร้างที่เหมาะสมสำหรับการแปล ซึ่งสอดคล้องกับแนวคิดของงานวิจัยนี้ ในการนำวิเทอร์บีอัลกอริทึมมาช่วยในการเลือกคำแปลและลำดับคำที่ถูกต้อง

ดังนั้น งานวิจัยนี้จึงขอนำเสนอ การลดความกำกวมในการแปลภาษาไทยเป็นภาษาไทยล้านนาโดยอิงพจนานุกรม ด้วยวิธีการของแบบจำลองฮิดเดนมาร์คอฟและวิเทอร์บีอัลกอริทึม ร่วมกับคลังคู่ประโยคไทย-ไทยล้านนา

ตัวอย่างความกำกวมในการแปลภาษาไทยเป็นภาษาไทยล้านนา เช่นในภาษาไทยคำว่า "บอก" จะพบคำแปลเป็นภาษาไทยล้านนาได้ 3 คำ คือ ᨠ᩵ᩣ᩠ᩅ (ก่าว), ᨲᩪ᩠ᨶ (ตูน) และ ᨷᩬᨠ (บอก)

ซึ่งแต่ละคำ จะถูกใช้ในสถานการณ์ที่แตกต่างกัน ขึ้นอยู่กับบริบทที่เกิดร่วมในประโยค คือ

"ᨻᩳ᩵ᩃ᩶ᩰ᩠ᨿᨦ ᨠ᩵ᩣ᩠ᩅ ᨸᩥᨯ ᨦᩣ᩠ᨶ" (พ่อเลี้ยงก่าวปิดงาน) ก่าว ใช้กับการพูดอย่างเป็นพิธีการ

"ᨻᩦ᩵ ᨲᩪ᩠ᨶ ᨧᩮᩢ᩶ᩣ" (พี่ตูนเจ้า) ตูน ใช้ในการพูดกับบุคคลระดับเจ้านาย(ฝ่ายเหนือ)

"ᩉᩖ᩵ᩡᩋ᩵ᩬ᩠ᨶ ᨷᩬᨠ ᨾᩯ᩵ " (หล่ะอ่อนบอกแม่) บอก ใช้ในการพูดในระดับสามัญชนทั่วไป

2. งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวกับการแปลภาษาไทยล้านนาในปัจจุบัน พบมีงานวิจัยทางด้านการแปลจากภาษาไทยล้านนาเป็นภาษาไทย [1] โดยใช้พจนานุกรมร่วมกับข่ายงานเพิ่มขยาย ซึ่งเป็นงานที่มีความใกล้เคียงกับงานวิจัยของเรา มีขั้นตอนดังภาพที่ 1

ภาพที่ 1: การแปลภาษาไทยล้านนาเป็นภาษาไทยกลาง [1]

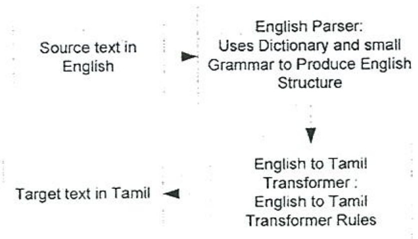
ขั้นตอนที่ 1 การตัดคำโดยใช้อัลกอริทึมแบบเลือกคำยาวสุด

ขั้นตอนที่ 2 ค้นหาคำแปลในพจนานุกรม ซึ่งมีการเก็บข้อมูลแบบทรัย

ขั้นตอนที่ 3 หากคำไหนไม่พบในพจนานุกรม จะใช้การปริวรรตภาษาไทยล้านนา แทนโครงสร้างพยางค์ไทยล้านนาด้วยออโตมาต้าเชิงไม่กำหนด และกำหนดให้ชนิดของคำเป็นคำนามชนิดวิสามายนาม

ขั้นตอนที่ 4 เลือกโครงสร้างประโยคไทยล้านนาและแจกแจงประโยคด้วยกฎโครงสร้างวลีและกฎโครงสร้างประโยคโดยใช้ข่ายงานเพิ่มขยาย

ขั้นตอนที่ 5 เลือกคำแปล โดยพิจารณาจากบริบทและการเกิดร่วมของคำ ในการเลือกคำแปลหลายๆ คำ



ภาพที่ 2: การแปลภาษาอังกฤษเป็นภาษาทมิฬ [2]

งานวิจัยแปลภาษาอังกฤษเป็นภาษาทมิฬ [2] ได้นำวิเทอร์บีอัลกอริทึมมาประยุกต์ใช้ ในการลดความกำกวมระดับโครงสร้างประโยค

งานวิจัยนี้มีกระบวนการทำงานหลักๆ อยู่ 2 กระบวนการ ดังภาพที่ 2 คือ

- กระบวนการเรียนรู้(Training phases) จะสร้างโมเดลทางสถิติสำหรับการแปล โดยใช้คลังข้อมูลภาษาอังกฤษภาษาทมิฬ
- กระบวนการแปล(Translation phases) จะใช้วิธีการค้นหาแบบฮิวริสติก เพื่อหาคำแปลที่ดีที่สุดของข้อความ

ลักษณะของภาษาทมิฬ เนื่องจากเป็นภาษาที่คำในภาษาทมิฬเกิดจากการรวมกันของหลาย ๆ หน่วยคำมีข้อมูลเพศและกริยาลงท้ายที่แตกต่างกัน ทำให้ต้องพิจารณาโดยบริบท ดังนั้นในกระบวนการแปลของงานวิจัยดังกล่าว จึงใช้แบบจำลองฮิดเด้นมาร์คอฟและวิเทอร์บีอัลกอริทึม ในการค้นหาโครงสร้างประโยคที่เหมาะสมสำหรับการแปลที่สุด มีขั้นตอนการทำงานดังนี้

ขั้นตอนที่ 1 คำนวณหาค่าความน่าจะเป็นของสถานะเริ่มต้น

ขั้นตอนที่ 2 คำนวณความน่าจะเป็นของสถานะถัดมาไปจนถึงสถานะสุดท้าย โดยที่สถานะข้างหน้า จะเป็นตัวกำหนดสถานะถัดมาที่น่าจะเป็นไปได้

ขั้นตอนที่ 3 ค้นหาเส้นทางที่มีความน่าจะเป็นสะสมสูงที่สุด

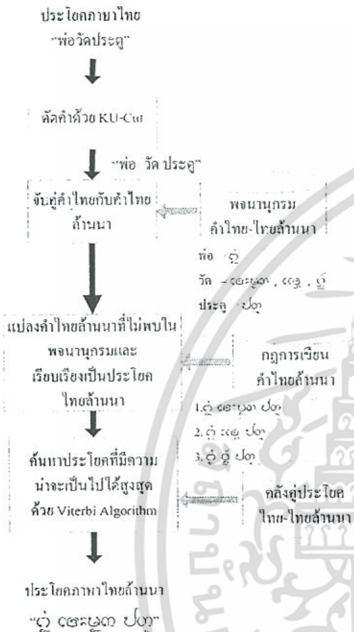
จากงานวิจัย [2] ทำให้ผู้วิจัยเห็นแนวทางการแก้ปัญหาความกำกวม ที่อาจเกิดขึ้นในการแปลภาษาไทยเป็นภาษาไทยล้านนา ด้วยการประยุกต์ใช้แบบจำลองของฮิดเดนมาร์คอฟและวิเทอร์บีอัลกอริทึม โดยอิงพจนานุกรม ในการลดความกำกวมสำหรับกรณีที่พบคำแปลภาษาไทยล้านนามากกว่า 1 คำ ซึ่งคำแปลที่เป็นผลลัพธ์จะได้มาจากคำแปลที่มีความน่าจะเป็นสูงสุด ทำให้การ

แปลภาษามีความใกล้เคียงกับธรรมชาติของภาษาล้านนามากยิ่งขึ้น

3. การลดความกำกวมในการแปลภาษาไทย-ไทยล้านนา

3.1 ภาพรวมของการแปลภาษาไทยเป็นภาษาไทยล้านนา

ในการแปลภาษาไทยเป็นภาษาไทยล้านนา โดยอิงพจนานุกรม มีขั้นตอนดังนี้



ภาพที่ 3: ภาพรวมของการแปลภาษาไทยเป็นภาษาไทยล้านนา

ขั้นตอนที่ 1 ตัดคำในประโยคภาษาไทยที่รับเข้ามา ด้วยโปรแกรม KU CUT พัฒนาโดยมหาวิทยาลัยเกษตรศาสตร์ ใช้เทคนิคการเรียนรู้จากข้อมูลแบบไม่ใช้ตัวอย่างร่วมกับการใช้พจนานุกรมและคำ [3] มีความถูกต้องของการตัดคำร้อยละ 79

ขั้นตอนที่ 2 การจับคู่คำไทยกับคำไทยล้านนา ขั้นตอนนี้จะนำคำไทยที่ตัดได้ไปค้นหาคำแปลในพจนานุกรมคำไทย-ไทยล้านนา ซึ่งคำไทยคำหนึ่งอาจจับคู่กับคำแปลไทยล้านนาได้หลายคำ จึงเป็นสาเหตุทำให้เกิดความกำกวมขึ้น

ขั้นตอนที่ 3 แปลงคำไทยล้านนาที่ไม่พบในพจนานุกรมและเรียบเรียงเป็นประโยคไทยล้านนา ขั้นตอนนี้จะนำคำไทยที่ไม่พบคำแปลในพจนานุกรมมาแปลงเป็นคำไทยล้านนาตามกฎการเรียงอักษรไทยล้านนาที่ได้มาจากผู้เชี่ยวชาญและการปริวรรตภาษาไทยล้านนา จากนั้นจะเรียบเรียงคำไทยล้านนาที่ได้จากขั้นตอนที่ 2 และ 3 เป็นประโยคไทยล้านนาที่เป็นไปได้ทั้งหมด

ขั้นตอนที่ 4 ค้นหาประโยคที่มีความน่าจะเป็นสูงสุด ด้วยวิเทอร์บิอัลกอริทึม เป็นขั้นตอนสำหรับลดความกำกวมของคำศัพท์ จะถูกใช้เมื่อประโยคไทยล้านนาที่เรียบเรียงได้ มีมากกว่า 1 ประโยค จะนำประโยคไปคำนวณหาประโยคที่มีความน่าจะเป็นสูงสุด ค่าความน่าจะเป็นที่จะนำมาคำนวณของแต่ละคำ ได้มาจากการนับความถี่ของคำในคลังคู่ประโยคไทย-ไทยล้านนา และผลลัพธ์สุดท้าย จะได้ประโยคไทยล้านนาที่มีค่าความน่าจะเป็นสะสมสูงสุด ซึ่งจะเป็นประโยคที่มีความเหมาะสมที่จะใช้ในการแปลที่สุด

3.2 การประยุกต์ใช้วิเทอร์บิอัลกอริทึมสำหรับการแปลภาษาไทยเป็นภาษาไทยล้านนา

แบบจำลองฮิดเดนมาร์คอฟ [4] เป็นเทคนิคการเรียนรู้ทางสถิติ (Statistical Machine Learning) ที่ใช้จำแนกรูปแบบของลำดับเหตุการณ์หนึ่งๆ ซึ่งลำดับของเหตุการณ์จะถูกจำลองให้ อยู่ในรูปแบบของลำดับการเปลี่ยนแปลงของสถานะ (State) จะมีค่าความน่าจะเป็นอยู่ 2 ชนิด คือ ค่าความน่าจะเป็นของการเปลี่ยนสถานะ และค่าความน่าจะเป็นของการทำให้เกิดผลลัพธ์ เมื่อมีการเปลี่ยนสถานะเกิดขึ้น และผลลัพธ์นี้จะถูกนำไปใช้ในการตัดสินใจว่าข้อมูลที่เข้ามามีโอกาสเป็นอะไรมากที่สุด

วิเทอร์บิอัลกอริทึม เป็นอัลกอริทึมหนึ่งในแบบจำลองฮิดเดนมาร์คอฟ ที่ใช้ในการจัดตำแหน่งลำดับของสถานะต่างๆ โดยลำดับของแต่ละสถานะ จะขึ้นอยู่กับความน่าจะเป็นสูงสุด ที่จะเป็นไปได้ทั้งหมด

การทำงานของวิเทอร์บิอัลกอริทึม [5] มี 3 ขั้นตอน คือ

ขั้นตอนที่ 1 คำนวณค่าความน่าจะเป็นของสถานะเริ่มต้นของข้อมูลนำเข้า

ขั้นตอนที่ 2 คำนวณความน่าจะเป็นของสถานะถัดไป ทุกสถานะ เพื่อจะเก็บเส้นทางที่ดีที่สุด สำหรับการเลือกเส้นทางในขั้นตอนถัดไป

ขั้นตอนที่ 3 การย้อนเส้นทาง เพื่อค้นหาลำดับความน่าจะเป็นสูงสุด จากสถานะที่ได้คำนวณไว้แล้ว ซึ่งจะได้ลำดับเส้นทางสถานะที่เป็นไปได้สูงสุดด้วย

การลดความกำกวมในการแปลภาษาไทยเป็นภาษาไทยล้านนาด้วยวิเทอร์บิอัลกอริทึม มีสมการและตัวแปรที่ต้องใช้ในอัลกอริทึม ดังต่อไปนี้

$$- \mathrm{PROB}(w_i|L_i) \cong \frac{\text{จำนวนการเกิดที่ } w_i \text{ มี } L_i \text{ เป็นคู่}}{\text{จำนวน } L_i \text{ ทั้งหมด}} \quad (1)$$

สมการที่ (1) ใช้ในการคำนวณหาความน่าจะเป็นที่ L_i จะเป็นคำคู่ของ w_i โดยที่ w_i คือ คำไทยที่ต้องการแปล และ L_i คือ คำแปลไทยล้านนาที่พบในพจนานุกรม

$$-\mathrm{PROB}(L_i|L_{i-1}) \cong \frac{\text{จำนวนการเกิดของ } L_i \text{ ตามหลัง } L_{i-1}}{\text{จำนวน } L_{i-1} \text{ ทั้งหมด}} \quad (2)$$

สมการที่ (2) ใช้ในการคำนวณหาความน่าจะเป็น เมื่อคำภาษาไทยล้านนา L_{i-1} จะตามมาด้วยคำไทยล้านนา L_i

- T คือ จำนวนคำในประโยคภาษาไทยที่ต้องการแปล
- N คือ จำนวนคำแปลภาษาไทยล้านนาที่พบในพจนานุกรมสำหรับคำไทยที่ต้องการแปล
- SEQSCORE คือ อาเรย์ขนาด N*T ใช้เก็บผลลัพธ์การคำนวณความน่าจะเป็นสูงสุด ของลำดับคำที่อาจจะแปลได้ทั้งหมด ตั้งแต่คำลำดับแรกจนถึงคำในลำดับปัจจุบัน
- BACKPTR คือ อาเรย์ขนาด N*T ใช้เก็บเส้นทางที่มีค่าความน่าจะเป็นสะสมสูงสุด
- C คือ ลิสต์ที่มีขนาด T ใช้เก็บชุดลำดับของคำไทยล้านนาที่มีค่าความน่าจะเป็นสูงสุด
- Con คือ ลิสต์ที่ใช้เก็บข้อมูลคำไทยที่จับคู่กับคำไทยล้านนาได้เพียง 1 คำ ลิสต์นี้มีประโยชน์จะช่วยลดจำนวนรอบในการคำนวณหาความน่าจะเป็น โดยไม่ต้องคำนวณทุกๆ คำไทยล้านนา

```
For i=1 to N do                                              (1)
    if w1 in Con do
        SEQSCORE(i,1)= PROB(w1 | L1)* PROB(L1 |ø)
        BACKPRT(i,1)=0
        Break;
    else
        SEQSCORE(i,1)= PROB(w1 | L1)* PROB(L1 |ø)
        BACKPRT(i,1)=0
-------------------------------------------------------------
For t=2 to T                                                 (2)
    For i=1 to N
        if    in Con do
            SEQSCORE(i,t)= Max j=1,N (SEQSCORE(j,t-1)
                           *PROB(Li | Lj))* PROB(wi | Li)
            BACKPRT(i,t)=index of j that gave the max above
            Break;
        else
            SEQSCORE(i,t)= Max j=1,N (SEQSCORE(j,t-1)
                           *PROB(Li | Lj))* PROB(wi | Li)
            BACKPRT(i,t)=index of j that gave the max above
-------------------------------------------------------------
C(T)=i that maximizes SEQSCORE(i,T)                          (3)
For i=T-1 to 1 do :   C(i)=BACKPTR(C(i+1),i+1)
```

ภาพที่ 4: วิเทอร์บีอัลกอริทึมในการลดความกำกวมในการแปลภาษาไทยเป็นภาษาไทยล้านนา

จากภาพที่ 4 วิเทอร์บีอัลกอริทึมที่ใช้ลดความกำกวมในการแปลภาษาไทยเป็นภาษาไทยล้านนา มีขั้นตอนดังนี้

ขั้นตอนที่ 1 คำนวณค่าความน่าจะเป็นของคำไทยล้านนาที่จะเป็นคำแปลของคำไทยคำแรก

ขั้นตอนที่ 2 คำนวณค่าความน่าจะเป็นของคำไทยล้านนาที่จะเป็นคำแปลของคำไทยคำถัดๆ ไปจนถึงคำสุดท้าย โดยจะเก็บตำแหน่งของคำไทยล้านนาที่มีความน่าจะเป็นสะสมสูงสุด ณ ตำแหน่งที่กำลังพิจารณาไว้ในลิสต์ BACKPTR เพื่อจดจำเส้นทาง

ขั้นตอนที่ 3 ย้อนเส้นทางในตัวแปร BACKPTR เพื่อหาลำดับคำไทยล้านนา ที่มีค่าความน่าจะเป็นสะสมสูงสุด จากการย้อนเส้นทางนี้จะทำให้ประโยคภาษาไทยล้านนา ที่มีค่าความน่าจะเป็นสะสมสูงสุดอยู่ในลิสต์ C

3.3 ตัวอย่างการลดความกำกวมในการแปลภาษาไทยเป็นภาษาไทยล้านนา

การแปลภาษาไทยเป็นภาษาไทยล้านนา ในที่นี้จะยกตัวอย่างโดยใช้คลังคู่ประโยคไทย-ไทยล้านนาจากนิทานพื้นบ้านเรื่อง "ปูกับนกยาง" ซึ่งมีจำนวนคู่ประโยค 29 ประโยค โดยมีประโยคตั้งต้นคือ "นก ยาง บอก" เมื่อผ่านขั้นตอนการตัดคำ และจับคู่คำศัพท์ สามารถเรียงเป็นประโยคไทยล้านนาได้ 6 ประโยคดังนี้

ประโยคที่ 1 "ຊົກ ພຊູ ຕູ່ວູ" (นก ญาง ก่าว)

ประโยคที่ 2 "ຊົກ ພຊູ ອູລ໌" (นก ญาง ตูน)

ประโยคที่ 3 "ຊົກ ພຊູ ບູດ" (นก ญาง บอก)

ประโยคที่ 4 "ສຕຸດ ພຊູ ຕູ່ວູ" (สะกุนะ ญาง ก่าว)

ประโยคที่ 5 "ສຕຸດ ພຊູ ອູລ໌" (สะกุนะ ญาง ตูน)

ประโยคที่ 6 "ສຕຸດ ພຊູ ບູດ" (สะกุนะ ญาง บอก)

ตารางที่ 1: แสดงค่าความถี่ของคำไทยคู่คำล้านนาจาก คลังคู่ประโยคไทย-ไทยล้านนา

คำไทยคู่คำล้านนา	ความถี่	คำไทยคู่คำล้านนา	ความถี่
นก/ຊົກ (นก)	11	บอก/ຕູ່ວູ (ก่าว)	0
นก/ສຕຸດ (สะกุนะ)	0	บอก/ອູລ໌ (ตูน)	0
ยาง/ພວດ (ญาง)	11	บอก/ບູດ (บอก)	2

ตารางที่ 2: แสดงค่าความถี่แบบยูนิแกรมของคำล้านนาจากคลังคู่ประโยคไทย-ไทยล้านนา

คำล้านนา	ความถี่	คู่คำล้านนา	ความถี่
ຊົດ (นก)	11	ຕູ່ວູ (ก่าว)	0
ສຕຸດ (สะกุนะ)	0	ອູລ໌ (ตูน)	0
ພຊູ (ญาง)	11	ບູດ (บอก)	2

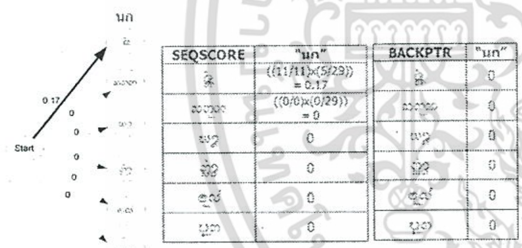
ตารางที่ 3: แสดงค่าความถี่แบบไบแกรมของคำล้านนาจากคลังคู่ประโยคไทย-ไทยล้านนา

คู่คำล้านนาที่ตามกันมา	ความถี่	คู่คำล้านนาที่ตามกันมา	ความถี่
ᨿᩣ᩠ᨦ | ᨶᩫ᩠ᨠ (ญาง|นก)	11	ᨯᩪ᩠ᨶ | ᨿᩣ᩠ᨦ (ดูน|ญาง)	0
ᨿᩣ᩠ᨦ | ᩈᨠᩩᨱ (ญาง|สะกุนะ)	0	ᨷᩬᨠ | ᨿᩣ᩠ᨦ (บอก|ญาง)	0
ᨴ᩵ᩣ᩠ᩅ | ᨿᩣ᩠ᨦ (เท่า|ญาง)	0	ᨶᩫ᩠ᨠ | Start (นก|เริ่มต้นประโยค)	5

ข้อสังเกต ค่าความถี่ของคำไทยคู่คำล้านนา และ ค่าความถี่แบบยูนิแกรมของคำล้านนา ของคำอื่นๆ ที่ไม่ได้ปรากฏในตารางที่ 1 ถึง ตารางที่ 3 ให้กำหนดมีค่าเท่ากับ 0 ส่วนค่าความถี่แบบไบแกรมของคำล้านนาของคำอื่นๆ ที่ไม่ได้ปรากฏในตารางที่ 1 ถึง ตารางที่ 3 ให้กำหนดมีค่าเท่ากับ 0.0001 [5]

จากนั้นนำประโยคทั้ง 6 มาผ่านกระบวนการลดความกำกวมด้วยวิเทอร์บีอัลกอริทึม ดังนี้

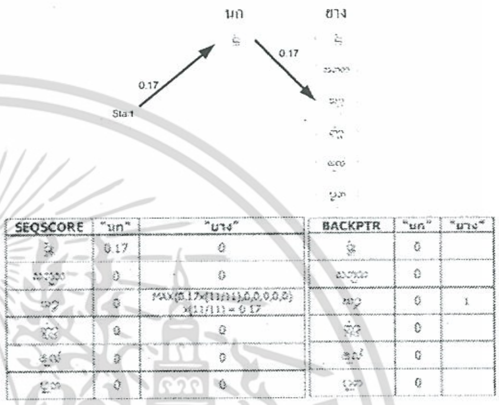
ขั้นตอนที่ 1 คำนวณค่าความน่าจะเป็นของคำไทยล้านนาที่จะเป็นคำแปลของคำไทยคำแรก ในที่นี้คือ คำว่า "นก" ตามการทำงานในกรอบที่ (1) ของภาพที่ 4



ภาพที่ 5: แสดงการคำนวณหาค่าความน่าจะเป็นของคำแปลของคำว่า "นก" ในตัวแปร SEQSCORE และ การเริ่มบันทึกเส้นทางในตัวแปร BACKPTR

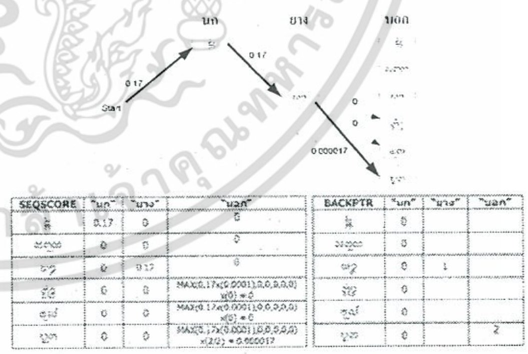
จากภาพที่ 5 ในตัวแปร SEQSCORE จะได้ค่าความน่าจะเป็นของคำว่า "ᨶᩫ᩠ᨠ" (นก) ที่จะเป็นคำแปลของคำไทย "นก" สูงที่สุดคือ 0.17 และ เริ่มการบันทึกเส้นทางลงในตัวแปร BACKPTR ในตำแหน่งคอลัมน์ของคำว่า "นก" กำหนดให้เป็น 0 เนื่องจากเป็นคำแรกของประโยค

ขั้นตอนที่ 2 คำนวณค่าความน่าจะเป็นของคำไทยล้านนาที่จะเป็นคำแปลของคำไทยคำถัดมา คือ คำว่า "ญาง" ตามการทำงานในกรอบที่ (2) ของภาพที่ 4 เนื่องจากในขั้นตอนของการจับคู่คำไทยกับคำไทยล้านนาคำว่า "ยาง" สามารถจับคู่กับคำล้านนาได้เพียง 1 คำ จึงมีการบันทึกตำแหน่งของคำไว้ในตัวแปร Con ซึ่งจะช่วยให้ไม่ต้องไปคำนวณค่าความน่าจะเป็นคำแปลของคำไทยล้านนาที่เหลืออื่น ๆ ที่มีในลิสต์ L ดังภาพที่ 6 แสดงการคำนวณหาค่าความน่าจะเป็นของคำว่า "ᨿᩣ᩠ᨦ" (ญาง) ที่จะเป็นคำแปลของคำว่า "ยาง" ในตัวแปร SEQSCORE โดยที่ความน่าจะเป็นที่ได้จะเป็นความน่าจะเป็นที่สะสม มาตั้งแต่คำเริ่มต้นมาจนถึงคำที่กำลังพิจารณา ซึ่งก็คือ คำว่า "ยาง" มีความน่าจะเป็นที่คำว่า "ᨿᩣ᩠ᨦ" (ญาง) จะเป็นคำแปลเท่ากับ 0.17 และทำการบันทึกเส้นทางเท่ากับ 1 ลงในตัวแปร BACKPTR



ภาพที่ 6: แสดงการคำนวณหาค่าความน่าจะเป็นของคำแปลของคำว่า "ยาง" ในตัวแปร SEQSCORE และ การบันทึกเส้นทางในตัวแปร BACKPTR

ต่อมาคำนวณหาค่าความน่าจะเป็นของคำไทยล้านนา ที่จะเป็นคำแปลของคำว่า "บอก"



ภาพที่ 7: แสดงการคำนวณหาค่าความน่าจะเป็นของคำแปลของคำว่า "บอก" ในตัวแปร SEQSCORE และ การบันทึกเส้นทางในตัวแปร BACKPTR

จากภาพที่ 7 ในตัวแปร SEQSCORE จะได้ค่าความน่าจะเป็นของคำว่า "ᨷᩬᨠ" (บอก) ที่จะเป็นคำแปลของคำไทย "บอก" สูงที่สุดคือ 0.000017 และ และทำการบันทึกเส้นทางเท่ากับ 2 ลงในตัวแปร BACKPTR

ขั้นตอนที่ 3 ย้อนเส้นทางเพื่อหาลำดับคำแปลไทยล้านนา ที่มีค่าความน่าจะเป็นสูงสุด จากลิสต์ BACKPTR ตามการทำงานในกรอบที่ (3) ของภาพที่ 4 จะทำให้ได้ประโยคภาษาไทยล้านนาที่มีค่าความน่าจะเป็นสะสมสูงสุดเป็นผลลัพธ์สุดท้ายอยู่ในตัวแปร C คือ ประโยค “ᨶᩫ᩠ᨠ ᨿᩣ᩠ᨦ ᨷᩬᨠ” (นก ญาง บอก)

4. การดำเนินงานในปัจจุบันและอนาคต

4.1 การดำเนินงานในปัจจุบัน

สร้างคลังคู่ประโยคไทย-ไทยล้านนาต้นแบบ จากนิทานเรื่อง “ปูกับนกกระยาง” ทำให้ได้คู่ประโยคไทย-ไทยล้านนา 29 คู่ ประกอบด้วยคำจำนวน 201 คำ โดยเป็นคำที่ไม่ซ้ำกันจำนวน 93 คำ และทดสอบการแปลภาษาไทยเป็นภาษาไทยล้านนา ด้วยประโยคทดสอบจำนวน 10 ประโยคที่ผู้วิจัยสร้างขึ้นเอง

วัดประสิทธิภาพการแปล โดยการคำนวณจากประโยคที่แปลถูกต้องต่อประโยคที่นำมาทดสอบทั้งหมด ดังสมการด้านล่าง

$$\text{ประสิทธิภาพ}(\%) = \frac{X}{N} \times 100 \quad (3)$$

ตัวแปร X คือ จำนวนของประโยคที่แปลถูกต้อง

ตัวแปร N คือ จำนวนของประโยคที่นำมาทดสอบทั้งหมด

4.1.1 ผลการทดลอง

แบ่งการทดลองเป็น 2 วิธี คือ วิธีที่ 1 ใช้วิเทอร์บีอัลกอริทึมช่วยลดความกำกวม ในกรณีที่คำไทยคำหนึ่งมีคำแปลไทยล้านนาได้หลายคำ เปรียบเทียบกับ วิธีที่ 2 ที่จะแก้ปัญหานี้โดยการเลือกคำแปลไทยล้านนาคำแรก ที่พบในพจนานุกรม ได้ผลการทดลองดังตารางที่ 4

ตารางที่ 4: ตารางเปรียบเทียบประสิทธิภาพของการแปล

วิธีการทดลอง	ผลลัพท์ที่แปลถูกต้อง
วิธีที่ 1	90%
วิธีที่ 2	30%

จากการทดลอง วิธีที่ 1 ให้ผลการแปลที่มีความถูกต้องมากกว่าวิธีที่ 2 ถึง 60% ดังนั้นจึงสรุปได้ว่าวิเทอร์บีอัลกอริทึมสามารถช่วยลดความกำกวมในกรณีที่คำไทยคำหนึ่งมีคำแปลไทยล้านนาหลายคำได้ และการแปลจะยิ่งมีประสิทธิภาพมากขึ้นเมื่อจำนวนคู่ประโยคในคลังคู่ประโยคไทย-ไทยล้านนามีมากขึ้น

4.2 การดำเนินงานในอนาคต

จะจัดสร้างคลังคู่ประโยคไทย-ไทยล้านนา 500 ประโยค โดยนำมาจากนิทานและวรรณกรรมพื้นบ้าน และใช้ประโยคทดสอบ 100 ประโยค โดยจะวัดประสิทธิภาพเปรียบเทียบกับความเห็นของผู้เชี่ยวชาญ 4 คน

5. สรุป

จากการทดลองเบื้องต้นวิเทอร์บีอัลกอริทึมสามารถช่วยลดความกำกวมในการแปลภาษาไทยเป็นภาษาไทยล้านนาได้ โดยเมื่อทดสอบกับประโยคที่สร้างจากคำไทยที่มีปรากฏในคลังคู่ประโยคไทย-ไทยล้านนา พบว่ามีประสิทธิภาพเพิ่มขึ้นร้อยละ 60 ในอนาคตผู้วิจัยจะรวบรวมประโยคในคลังคู่ประโยคไทย-ไทยล้านนาให้ได้มากขึ้น เพื่อรองรับการแปลภาษาไทยเป็นภาษาไทยล้านนาให้ดียิ่งขึ้น

เอกสารอ้างอิง

[1] พุธยดี ศิริแสงตระกูล และ นิกร ยาพรม, “การแปลภาษาล้านนาเป็นภาษาไทยกลาง”, *วารสารวิจัย มหาวิทยาลัยขอนแก่น*, มีนาคม 2552 หน้า 264-274.

[2] S.Vetrivel and Diana Baby, “English to Tamil Statistical Machine Translation and Alignment Using HMM”, *Recent Advances in Networking, VLSI and Signal Processing* University of Cambridge, UK, 20-22 February 2010, pp.182-186.

[3] สุธี สุดประเสริฐ และ อัศนีย์ ก่อตระกูล, “การตัดคำภาษาไทยด้วยเทคนิคการเรียนรู้แบบไม่ใช้ตัวอย่าง”, *The 7th National Computer Science and Engineering Conference (NCSEC'2003)*, Chonburi Thailand, 2003.

[4] ยศวัรธน์ แสนสิงห์ และ อัครา ประโยชน์, “การรู้จำชื่อเฉพาะภาษาไทยโดยใช้แบบจำลองฮิดเดนมาร์คอฟ”, *Proceedings of the Second Conference on Knowledge and Smart Technologies 2010 (KST-2010)*, Chonburi Thailand, 24-25 July 2010.

[5] James Allen, *Natural Language Understanding*, The Benjamin/Cummings Publishing Company, Inc.:Redwood City, CA, 1995.

nccit 2012

THE 8TH NATIONAL CONFERENCE ON COMPUTING AND INFORMATION TECHNOLOGY

11-12 MAY 2012

FACULTY OF INFORMATION TECHNOLOGY
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK (KMUTNB)

WWW.NCCIT.NET

ประวัติผู้เขียน

ชื่อ-นามสกุล นายประธาน คำจินะ

วัน เดือน ปีเกิด วันที่ 20 เมษายน 2526

ประวัติการศึกษา

- ระดับประถมศึกษา โรงเรียนวัดท่าโป่ง
- ระดับมัธยมศึกษาตอนต้น โรงเรียนสันป่าตองวิทยาคม
- ระดับมัธยมศึกษาตอนปลาย โรงเรียนสันป่าตองวิทยาคม
- ระดับอุดมศึกษา วิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยราชภัฏเชียงใหม่

ประวัติการทำงาน

- พ.ศ. 2548 - 2550 พนักงานมหาวิทยาลัย ตำแหน่ง นักวิชาการคอมพิวเตอร์ สำนักส่งเสริมวิชาการ มหาวิทยาลัยราชภัฏเชียงใหม่
- พ.ศ. 2550 - ปัจจุบัน พนักงานมหาวิทยาลัยสายวิชาการ สาขาวิชาคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏเชียงใหม่