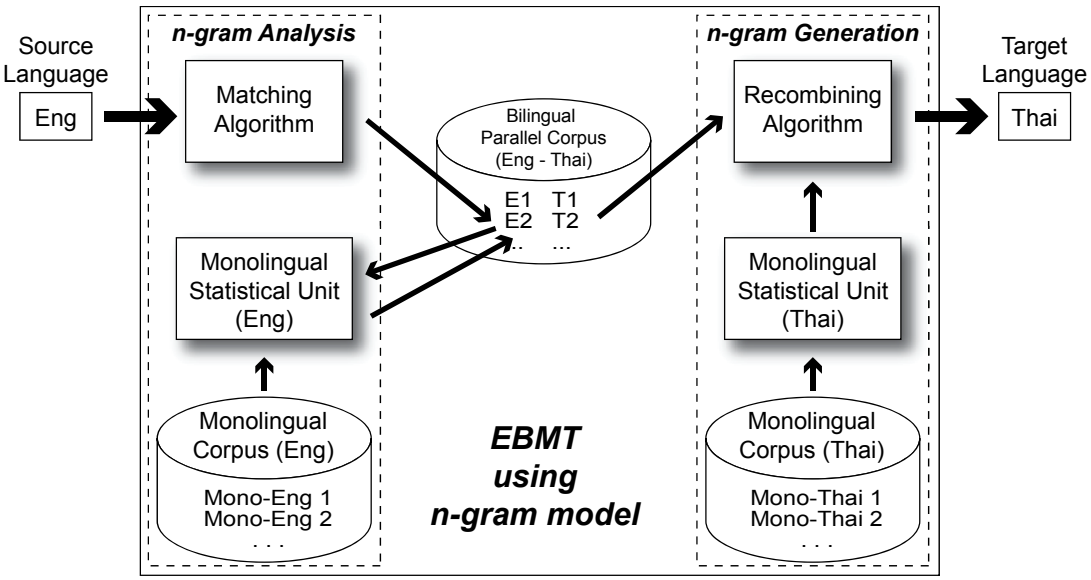
บทที่ 3
ระบบแปลภาษาอังกฤษ-ไทยด้วยเครื่องแบบอิงตัวอย่าง โดยใช้ตัวแบบเอ็นแกรม

ในบทนี้จะกล่าวถึงรายละเอียดต่างๆ ของระบบแปลภาษาอังกฤษ-ไทยด้วยเครื่องแบบอิงตัวอย่างโดยใช้ตัวแบบเอ็นแกรม (ขอเรียกโดยย่อว่า “ระบบ”) อันประกอบด้วย สถาปัตยกรรมระบบ (System Architecture) แนวคิดของเอ็นแกรม (The *n*-gram Approach) ส่วนโปรแกรมการวิเคราะห์แบบเอ็นแกรม (*n*-gram analysis Component) ส่วนโปรแกรมการก่อกำเนิดแบบเอ็นแกรม (*n*-gram generation Component) และคลังข้อความ (Corpus)

3.1 สถาปัตยกรรมระบบ (System Architecture)

สถาปัตยกรรมของระบบนี้ได้ถูกแสดงไว้ในภาพที่ 3.1 ระบบนี้ประกอบด้วยส่วนสำคัญ 2 ส่วน คือ ส่วนโปรแกรมการวิเคราะห์แบบเอ็นแกรม (*n*-gram analysis component) และ ส่วนโปรแกรมการก่อกำเนิดแบบเอ็นแกรม (*n*-gram generation component) ระบบนี้ได้นำแนวคิดแบบตัวแบบเอ็นแกรมเข้าแก้ปัญหาความหลากหลายของแบบอย่าง (pattern) โดยอิงจากประโยคหรือส่วนของประโยค เมื่อระบบได้รับข้อมูลประโยคภาษาต้นทาง จะส่งข้อมูลดังกล่าวเพื่อทำการหาประโยคที่เข้าคู่ (match) กันได้ในคลังข้อความแบบคู่ ทุกประโยคหรือส่วนของประโยคที่เข้าคู่จะถูกนำไปเปรียบเทียบเพื่อหาผลลัพธ์ที่มีความใกล้เคียงกับคลังข้อความแบบเดี่ยวสำหรับภาษาปลายทาง

คลังข้อความแบบเดี่ยวถูกใช้ในขั้นตอนวิเคราะห์และก่อกำเนิดเพื่อหาผลลัพธ์ที่ดีที่สุดสำหรับแต่ละประโยคหรือส่วนของประโยคที่เข้าคู่กัน ขณะที่คลังข้อความแบบคู่จะประกอบด้วยคู่ประโยคหรือส่วนของประโยคของภาษาอังกฤษและภาษาไทยจำนวนมาก โดยข้อมูลในคลังข้อความแบบคู่จะทำหน้าที่เป็นกฎการโอนย้าย (Transfer Rule) อย่างไรก็ดี คลังข้อความทั้งหมดที่ใช้ในวิทยานิพนธ์นี้ถูกเตรียมโดยนักภาษาศาสตร์ของแผนกเทคโนโลยีประมวลผลข้อความ ฝ่ายวิจัยและพัฒนาเทคโนโลยีสารสนเทศ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ ประเทศไทย (NECTEC)



ภาพที่ 3.1 ระบบแปลภาษาอังกฤษ-ไทยด้วยเครื่องแบบอิงตัวอย่างโดยใช้ตัวแบบเอ็นแกรม

3.2 แนวคิดของเอ็นแกรม (The *n*-gram Approach)

ตัวแบบภาษาของเอ็นแกรม (*n*-gram language model) อยู่ภายใต้สมมติฐานต่อไปนี้ คำตำแหน่งที่ n จะมีความสัมพันธ์เฉพาะกับตัวก่อนหน้าของคำนั้นๆ ซึ่งก็คือคำตำแหน่งที่ n-1 ดังนั้น ค่าประมาณความน่าจะเป็นของตัวแบบ $P(w)$ จะสามารถเขียนได้เป็น $P(w_n \mid w_1,...,w_{n-1})$ สำหรับประโยคที่มีจำนวน N คำ

ให้ $w_1, w_2,..., w_N$ เป็นคำที่มาจากคลังข้อความแบบคู่ จะสามารถคำนวณความน่าจะเป็นของทั้งประโยคได้ดังสมการต่อไปนี้

$$P(w) = \sum_{i=1}^{N} P(w_i \mid w_{i-n+1},..., w_{i-1}) \tag{3.1}$$

หากคลังความมีขนาดใหญ่เพียงพอ ความน่าจะเป็นของ $P(w) = \sum_{i=1}^{N} P(w_i \mid w_{i-n+1},..., w_{i-1})$ จะคำนวณโดยหลักการความควรจะเป็นสูงสุด (Maximum likelihood principle) ได้เป็น

$$P(w_n \mid w_1,..., w_{n-1}) = \frac{C(w_1,..., w_n)}{C(w_1,..., w_{n-1})} \tag{3.2}$$

เมื่อ $C(w_1,..., w_{n-1})$ และ $C(w_1,..., w_n)$ คือ จำนวนครั้งที่ปรากฏของคำในสายข้อความ $w_1,..., w_{n-1}$ และ $w_1,..., w_n$ ตามลำดับ

3.3 ส่วนโปรแกรมการวิเคราะห์แบบเอ็นแกรม (*n*-gram analysis Component)

ส่วนโปรแกรมการวิเคราะห์แบบเอ็นแกรมมีจุดมุ่งหมายเพื่อวิเคราะห์ประโยคต้นแบบโดยตัดส่วนของประโยคออกมาในทุกๆ ทางที่เป็นไปได้ และเข้าคู่ตัวเลือกที่เหมาะสมที่สุด การทำงานหลักของส่วนโปรแกรมนี้คือ ขั้นตอนวิธีเข้าคู่ (Matching Algorithm)

ขั้นตอนวิธีเข้าคู่ จะถูกนำมาใช้เพื่อค้นหาประโยคหรือส่วนของประโยคที่ยาวที่สุด ส่วนของประโยคที่ยาวที่สุดจะอนุมานจากการเข้าคู่ของข้อมูลในคลังข้อความคู่ภาษา (คลังข้อความแบบคู่) ในกรณีที่มีทางให้เลือกมากกว่า 1 ทาง ระบบจะเลือกส่วนของประโยคที่สมควรนำมาใช้โดยอิงจากข้อมูลทางสถิติ ที่ได้จากคลังข้อความแบบเดี่ยวสำหรับภาษาต้นทาง

ส่วนโปรแกรมการวิเคราะห์แบบเอ็นแกรมมีขั้นตอนการทำงานดังนี้

Step 1: Define *Fragment_Set = {SL}* and
Result_Set = { }
Step 2: Generate sub-fragment Sf_i from *Fragment_Set* by segmenting groups of words that $next(w_i) \neq w_{i+1}$
Step 3: For each Sf_i that has more than one elements, find the maximum sub-sentence Max_Sf_i in Sf_i
Step 3.1: Push Max_Sf_i into *Result_Set*
Step 3.2: Delete Max_Sf_i from *Fragment_Set*
Step 4: Repeat Step 1 until each sub-fragment Sf_i has only one element
Step 5: Return Sf_i
Step 6: Return *Result_Set*

จากประโยคตัวอย่าง “Arsenal picked up a big victory in Champions League” เราตั้งสมมติฐานว่า มีส่วนของประโยค {a big victory}, {picked up}, และ {Champions League} อยู่ในคลังข้อความคู่ภาษา ทำให้ Matching Algorithm จะประมวลผลตามลำดับขั้นตอนต่อไปนี้

Step 1: *Fragment_Set = { {Arsenal picked up a big victory in Champions League} } ,*
Result_Set = { }
Step 2: Since $next(w_i) = w_{i+1}$ for all w_i , sub-fragment Sf_i has only one sub-fragment that is
Sf_1 *= {Arsenal picked up a big victory in Champions League}*
Step 3: maximum sub-sentence
Max_sf_1 *= {a big victory}*
Step 3.1: *Result_Set ={ {a big victory} }*
Step 3.2: *Fragment_Set = { {Arsenal picked up},*
{in Champions League} }
Step 4: *Fragment_Set = { {Arsenal picked up},*
{in Champions League} } ,
Result_Set = { {a big victory} }
Step 5: Since $next(w_i) \neq w_{i+1}$ at w_i= up, there are two sub-fragments,
Sf_1 *= {Arsenal picked up} and*
Sf_2 *= {in Champions League}*
Step 6: maximum sub-sentence Max_sf_1 *= {picked up}*
Step 6.1: *Result_Set ={ {picked up}, {a big victory} }*
Step 6.2: *Fragment_Set = { {Arsenal}, {in Champions League} }*
Step 7: maximum sub-sentence Max_sf_2 *= {in Champions League}*

Step 7.1: *Result_Set = { {Champions League}, {picked up}, {a big victory} }*
Step 7.2: *Fragment_Set = { {Arsenal}, {in} }*
Step 8: Since $next(w_i) \neq w_{i+1}$ at w_i= Arsenal, there are two sub-fragments Sf_1 = *{ Arsenal }* and Sf_2 =*{ in }*
Step 9: Return *Result_Set = { {Arsenal}, {in}, {Champions League}, {picked up}, {a big victory} }*

ผลจากส่วนโปรแกรมการวิเคราะห์แบบเอ็นแกรมจะถูกแปลเป็นภาษาไทยบนพื้นฐานของเกณฑ์ 2 เกณฑ์ ถ้าหากมีสมาชิกใดๆ ใน Result_Set ที่ไม่ใช่คำโดด ให้ใช้คำแปลจากคลังข้อความคู่ภาษาถ้าหากสมาชิกใน Result_Set เป็นคำโดด ให้ดึงคำแปลมาจากพจนานุกรมแทน ผลลัพธ์การแปลจาก Result_Set จะถูกส่งไปที่ส่วนโปรแกรมการก่อกำเนิดแบบเอ็นแกรม เช่น Result_Set = {{Arsenal}, {in}, {Champions League}, {picked up}, {a big victory}} จะถูกแปลเป็น {{เจ้าปืนใหญ่อาร์เซนอล}, {ใน}, {แชมเปี้ยนลีค}, {ได้}, {ชัยชนะครั้งใหญ่}}

3.4 ส่วนโปรแกรมการก่อกำเนิดแบบเอ็นแกรม (*n*-gram generation Component)

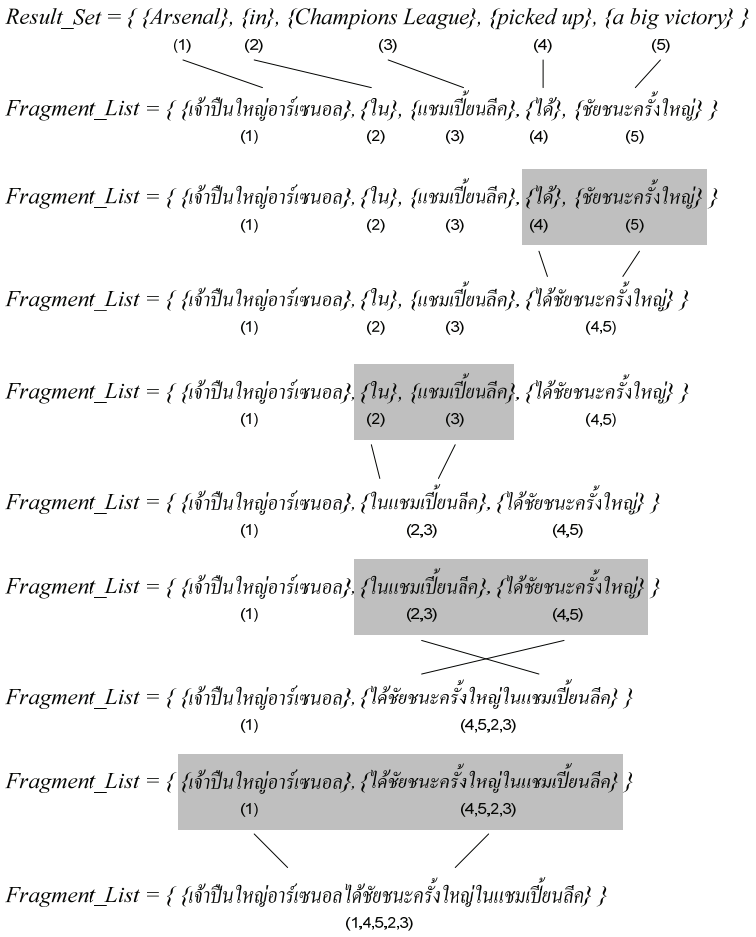
ส่วนโปรแกรมการก่อกำเนิดแบบเอ็นแกรมมีไว้เพื่อสร้างประโยคในภาษาปลายทาง โดยการรวมและเรียงลำดับชิ้นส่วนของประโยคให้เป็นประโยคเต็ม หน้าที่หลักของส่วนนี้คืออัลกอริธึมการ Recombine ในส่วนนี้จะใช้ Greedy Algorithm มาตรวจจับส่วนของประโยคที่ดีที่สุดที่สามารถนำมาต่อกันได้ กระบวนการ Recombine จะนำส่วนของประโยคมารวมกันเป็นประโยคในภาษาปลายทางโดยพิจารณาตามลำดับในประโยค แต่ละส่วนของประโยคจะนำมาต่อกันกับตัวใกล้เคียงซึ่งมีความถี่ที่ปรากฏในคลังข้อความสูงที่สุด กระบวนการ Recombine จะมีขั้นตอนดังต่อไปนี้

Step 1: Define *Fragment_List* = $\{Fr_1, Fr_2, \ldots, Fr_n\}$
Step 2: [combine the maximum probability of sub-sentence and its neighbor]
For each Fr_a and Fr_b in *Fragment_List* that
$1 \leq a, b \leq n$ and $|a-b| = 1$, Fr_{ab} = max_combine (Fr_a , Fr_b)
Step 3: Substitute Fr_a and Fr_b with Fr_{ab} and
delete Fr_a and Fr_b from *Fragment_List*
Fragment_List = $\{Fr_1, Fr_2, \ldots, Fr_{ab}, \ldots, Fr_n\}$
Step 4: Repeat Step 1 until *Fragment_List* has only one element
Step 5: Return *Fragment_List*

จากตัวอย่างข้างบน ผลลัพธ์ Result_Set ในส่วนวิเคราะห์โครงสร้างประโยคด้วย *n*-gram จะได้เป็น {{Arsenal}, {in}, {Champions League}, {picked up}, {a big victory}} ซึ่งสามารถนำมาแปลเป็นรายคำได้เป็น Fragment_List = { เจ้าปืนใหญ่อาร์เซนอล, ใน, แชมเปี้ยนลีค, ได้, ชัยชนะครั้งใหญ่ }

เมื่อนำกระบวนการ Recombine มาใช้ ส่วนของประโยคใน Fragment_List จะถูกรวมในขั้นตอนที่ 2 โดยพิจารณาจากส่วนของประโยคและส่วนใกล้เคียง ณ ระยะกระจัด 1 ดังได้แสดงไว้

ในภาพที่ 2 ส่วนที่ทาบสีเทาเอาไว้จะเป็นคู่ของคำหรือวลีที่สามารถนำมา combine กันได้ในขั้นตอนต่อไป ในท้ายที่สุดแล้ว จะสังเคราะห์ผลการแปลได้เป็นประโยค “เจ้าปืนใหญ่อาร์เซนอลได้ชัยชนะครั้งใหญ่ในแชมเปี้ยนลีค” ออกมาดังภาพที่ 3.2



ภาพที่ 3.2 ตัวอย่างการทำงานส่วนโปรแกรมการก่อกำเนิดแบบเอ็นแกรม

3.5 คลังข้อความ (Corpus)

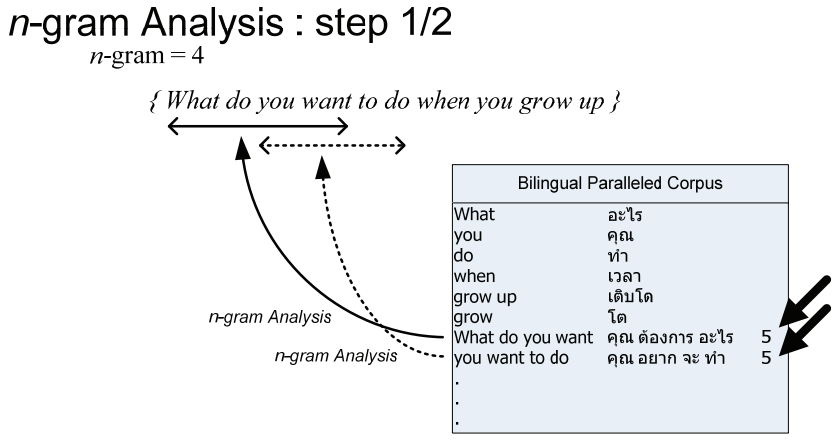
คลังข้อความที่ใช้ในระบบประกอบคลังข้อความหลายประเภท ได้แก่ คลังข้อความแบบเดี่ยว (Monolingual Corpus) และ คลังข้อความแบบคู่ (Bilingual Paralleled Corpus) เนื่องจากระบบนี้เป็นแบบออฟไลน์ ดังนั้นคลังข้อความทุกประเภทจะต้องถูกจัดเตรียมไว้ก่อนแล้ว

คลังข้อความจะถูกจัดเตรียมโดยนำส่วนที่เป็นภาษาไทยไปผ่านกระบวนการตัดคำซึ่งเป็นความสามารถที่มีอยู่แล้วใน Java 2 Platform, Standard Edition (J2SE) 5.0 จาก จากนั้นจึงนำไปเข้าสู่กระบวนการเรียนรู้ค่าสถิติของตัวแบบเอ็นแกรม

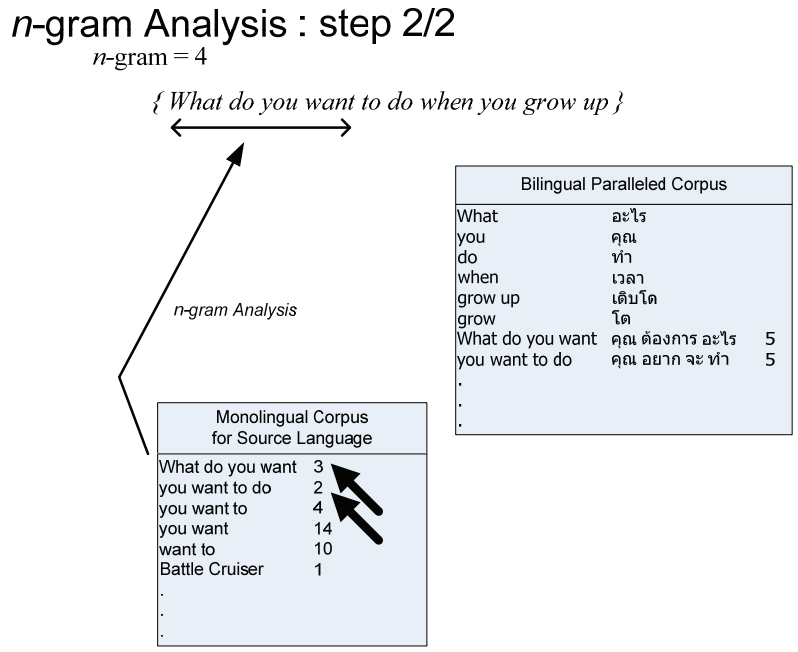
3.5.1 คลังข้อความแบบเดี่ยว (Monolingual Corpus)

คลังข้อความแบบเดี่ยวมี 2 ประเภท คือ คลังข้อความแบบเดี่ยวสำหรับภาษาต้นทาง (Monolingual Corpus for Source Language) และ คลังข้อความแบบเดี่ยวสำหรับภาษาปลายทาง (Monolingual Corpus for Target Language) โดยคลังข้อความทั้งสองประเภทเกิดจากการคำนวณค่าความถี่แบบเอ็นแกรม (*n*-gram frequency) ไว้ล่วงหน้า โดยจะมีการเตรียมความถี่ของ 2-gram, 3-gram และ 4-gram [15] และถูกจัดเก็บไว้ในรูปแบบแฟ้มข้อมูลแบบคั่นด้วยอักขระตั้งระยะ (TSV : Tab-separated value) โดยจะมีทั้งหมด 2 สดมภ์ (column) สดมภ์ที่หนึ่งจะเป็นข้อความที่ถูกแบ่งคำไว้แล้ว โดยจำนวนคำที่ถูกแบ่งจะหมายถึงจำนวนเอ็นแกรมของระเบียน (record) สดมภ์ที่สองคือจำนวนความถี่แบบเอ็นแกรม โดยสดมภ์ที่สองนี้จะมีหรือไม่ก็ได้ หากไม่มีจะถือว่ามีค่าความถี่แบบเอ็นแกรมเป็นหนึ่ง

คลังข้อความแบบเดี่ยวสำหรับภาษาต้นทาง (Source Language) ถูกใช้ในขั้นตอนส่วนโปรแกรมการวิเคราะห์แบบเอ็นแกรมเพื่อแก้ปัญหาความกำกวมในการหาส่วนของประโยคสำหรับภาษาต้นทาง โดยความกำกวมในการหาส่วนของประโยคสำหรับภาษาต้นทางจะเกิดขึ้นในขณะที่ส่วนโปรแกรมการวิเคราะห์แบบเอ็นแกรมพบว่า มีจำนวนส่วนของประโยคที่สามารถเข้าคู่กันได้และมีความยาวเท่ากันมากกว่า 1 คู่ โดยส่วนของประโยคที่พบดังกล่าวมีการซ้อนทับกัน (overlap) ในกรณีเช่นนี้ ระบบจะเลือกส่วนของประโยคที่มีการซ้อนทับกันซึ่งมีค่าความถี่สูงสุดจากคลังข้อความแบบคู่ แต่ก็เป็นไปได้ที่ความถี่ของส่วนของประโยคที่เข้าคู่และมีการซ้อนทับกันจะมีค่าความถี่เท่ากัน หากเป็นเช่นนี้ ระบบจะนำทุกประโยคต้นทางจากส่วนของประโยคที่เข้าคู่และซ้อนทับกันทั้งหมดที่ตรวจพบ นำไปเปรียบเทียบเพื่อหาส่วนของประโยคที่มีความถี่สูงสุดในคลังข้อความแบบเดี่ยวสำหรับภาษาต้นทาง หากความถี่ในคลังข้อความแบบเดี่ยวสำหรับภาษาต้นทางยังเท่ากันอยู่อีก ระบบจะเลือกส่วนของประโยคที่เข้าคู่และซ้อนทับกันประโยคแรกโดยอิงตำแหน่งเริ่มต้นจากซ้ายไปขวา



ภาพที่ 3.3 การแก้ปัญหาความกำกวม (1/2)



ภาพที่ 3.4 การแก้ปัญหาความกำกวม (2/2)

คลังข้อความแบบเดี่ยวสำหรับภาษาต้นทางจะถูกนำไปคำนวณเป็นสถิติสำหรับคลังข้อความแบบเดี่ยวสำหรับภาษาต้นทาง (Monolingual Statistical Unit for Source Language) โดยจะคำนวณหาค่าความถี่ของเอ็นแกรม (*n*-gram frequency) ตั้งแต่ 2-gram ถึง 4-gram จากข้อความภาษาต้นทาง (ภาษาอังกฤษ) ซึ่งสามารถรวบรวมได้โดยง่ายจาก webpage ทั่วไปที่มีเฉพาะภาษาต้นทางเท่านั้น ในงานวิจัยนี้ได้ทำการรวบรวมค่าความถี่ของเอ็นแกรมจากทั้งหมด 4,000 webpage โดยแบ่งได้เป็น 104,893 ประโยค หรือ 561,387 คำ

What do you want	3
you want to do	2
you want to	4
you want	14
want to	10
.	
.	
.	

ภาพที่ 3.5 ตัวอย่างคลังข้อความแบบเดี่ยวสำหรับภาษาต้นทาง

คลังความแบบเดี่ยวสำหรับภาษาปลายทาง (Target Language) ถูกใช้ในขั้นตอนส่วนโปรแกรมการก่อกำเนิดแบบเอ็นแกรมเพื่อใช้ตัดสินลำดับที่ถูกต้องในกระบวนการ Recombine เพื่อสร้างประโยคผลลัพธ์ในภาษาปลายทางที่ถูกต้อง

เป็น อันตราย	5
เป็น อันตราย ต่อ	2
เป็น อันตราย ถึง	1
ก็ เป็น อันตราย	1
อาจ เป็น อันตราย	1
ซึ่ง อาจ เป็น อันตราย	1
.	
.	
.	

ภาพที่ 3.6 ตัวอย่างคลังข้อความแบบเดี่ยวสำหรับภาษาปลายทาง

ในงานวิจัยชิ้นนี้ คลังข้อความแบบเดี่ยวสำหรับภาษาปลายทางจะถูกนำไปคำนวณเป็นสถิติสำหรับคลังข้อความแบบเดี่ยวสำหรับภาษาปลายทาง (Monolingual Statistical Unit for Target Language) โดยจะคำนวณหาค่าความถี่ของเอ็นแกรม (*n*-gram frequency) ตั้งแต่ 2-gram ถึง 4-gram ไว้ล่วงหน้า จากแหล่งข้อความภาษาปลายทางต่อไปนี้

- คลังข้อความทั่วไปเฉพาะภาษาปลายทางที่ถูกแบ่งประโยคและแบ่งคำไว้แล้ว มีจำนวน 13,758 ประโยค และมีจำนวนคำทั้งหมด 152,396 คำ
- คลังข้อความข่าวหนังสือพิมพ์ไทยรัฐ ย้อนหลัง 10 ปี ซึ่งถูกแบ่งประโยคและแบ่งคำไว้แล้ว มีจำนวน 138,812 ประโยค และมีจำนวนคำทั้งหมด 10,011,970 คำ

3.5.2 คลังข้อความแบบคู่ (Bilingual Paralleled Corpus)

คลังข้อความแบบคู่ ทำหน้าที่เป็นข้อมูลตัวอย่างประโยค ซึ่งเป็นหัวใจหลักสำหรับระบบแปลภาษาด้วยเครื่องแบบอิงตัวอย่าง คลังข้อความแบบคู่นี้ส่งผลโดยตรงต่อประสิทธิภาพการแปลของระบบ หากคลังข้อความแบบคู่มีข้อมูลตัวอย่างมากเพียงพอและถูกต้อง ข้อมูลตัวอย่างควรมีลักษณะเป็นส่วนของประโยคที่ปรากฏซ้ำๆ กันและมีปริมาณมาก คู่ความหมายควรมีความหมายที่เป็นกลางไม่เฉพาะเจาะจงมากนัก ผลการแปลจึงจะมีประสิทธิภาพและมีความแม่นยำสูง

What	อะไร	
you	คุณ	
do	ทำ	
when	เวลา	
grow up	เติบโต	
grow	โต	
What do you want	คุณ ต้องการ อะไร	5
you want to do	คุณ อยาก จะ ทำ	5
.		
.		
.		

ภาพที่ 3.7 ตัวอย่างคลังข้อความแบบคู่

หากคลังข้อความแบบคู่ มีการจัดเก็บคู่ภาษาต้นทางในระดับคำ จะเห็นได้อย่างชัดเจนว่าคลังข้อความแบบคู่จะมีสภาพไม่แตกต่างไปจากพจนานุกรมสองภาษา ดังนั้นพจนานุกรมสองภาษาจึงเสมือนว่าถูกรวมอยู่ในคลังข้อความแบบคู่ด้วย ซึ่งทำให้ส่วนโปรแกรมการวิเคราะห์แบบเอ็นแกรมสามารถหาความหมายของคำเมื่อ *n=1* ได้สำเร็จ

คลังข้อความแบบคู่นี้ถูกจัดเก็บไว้ในรูปแบบแฟ้มข้อมูลแบบคั่นด้วยอักขระตั้งระยะ โดยจะมีทั้งหมด 3 สดมภ์ สดมภ์ที่หนึ่งจะเป็นข้อความสำหรับภาษาต้นทางที่ถูกแบ่งคำไว้แล้ว โดยจำนวนคำที่ถูกแบ่งจะหมายถึงจำนวนเอ็นแกรมของระเบียน สดมภ์ที่สองจะเป็นข้อความสำหรับภาษาปลายที่ถูกแบ่งคำไว้แล้ว โดยข้อมูลในสดมภ์ที่สองนี้จะถูกใช้เป็นคำแปลของภาษาปลายทาง สดมภ์ที่สามคือจำนวนความถี่แบบเอ็นแกรมที่แสดงถึงอัตราการปรากฏขึ้นระหว่างคู่ความหมายของภาษาต้นทางและภาษาปลายทาง ทั้งนี้สดมภ์ที่สามนี้จะมีหรือไม่ก็ได้ หากไม่มีจะถือว่ามีจำนวนความถี่แบบเอ็นแกรมเป็นหนึ่งโดยปริยาย

ในงานวิจัยชิ้นนี้ คลังข้อความแบบคู่จะถูกคำนวณค่าความถี่ของเอ็นแกรม (*n*-gram frequency) ตั้งแต่ 2-gram ถึง 4-gram ไว้ล่วงหน้า จากแหล่งคลังข้อความต่อไปนี้

- ตัวอย่างประโยคจากหนังสือ English by Examples จำนวน 100,804 ประโยค ซึ่งถูกแบ่งประโยคและแบ่งคำไว้แล้ว
- พจนานุกรมอังกฤษ-ไทย Lexitron จำนวน 97,791 คำ ซึ่งถูกแบ่งคำไว้แล้ว
- พจนานุกรมไทย-อังกฤษ Lexitron จำนวน 104,892 คำ ซึ่งถูกแบ่งคำไว้แล้ว

เนื่องจากคลังข้อความแบบคู่มักหาไม่ได้โดยทั่วไปนัก อีกทั้งยังรวบรวมได้ยากเนื่องจากระบบต้องการคู่ความหมายระดับประโยคหรือส่วนของประโยค แต่แหล่งข้อมูลส่วนมากแม้จะมีลักษณะเป็นคู่ภาษากันอยู่แล้ว แต่ก็ยังไม่ได้แบ่งแบบประโยคต่อประโยค (โดยมากจะพบแบบย่อหน้าต่อย่อหน้า) อย่างไรก็ดีระบบมีความจำเป็นต้องใช้คลังข้อความแบบคู่เป็นปริมาณมากพอเพื่อให้ประสิทธิภาพในการแปลออกมาดี จำเป็นต้องรวบรวมคลังข้อความแบบคู่เพิ่มเติมให้ได้มากที่สุด