หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

แนวคิดและทฤษฎี

มนุษย์สามารถติดต่อสื่อสารถึงกันได้โดยใช้ภาษาที่เหมือนกัน ในการถ่ายทอดความรู้สึกความเข้าใจ โดยการพูดคุย การเขียนหรือการอ่าน ภาษาที่มนุษย์ใช้นี้นักคอมพิวเตอร์เรียกว่า "ภาษาธรรมชาติ"(Natural Language) (ยืน ภู่วรวรรณและคณะ , 2535)

โดยหลักการสื่อสารดังกล่าว เมื่อมนุษย์จะติดต่อสื่อสารกับคอมพิวเตอร์นั้น จำเป็นจะต้องใช้ภาษาสื่อสารที่คอมพิวเตอร์จะสามารถรับรู้ และทำความเข้าใจได้เท่านั้น นั่นคือมนุษย์จะต้องรู้จักใช้ภาษาคอมพิวเตอร์ ซึ่งมีอยู่มากมาย เช่น ภาษาเครื่อง (Machine Language) ภาษาแอสเซมบลี (Assembly Language) ภาษาระดับสูง และโปรแกรมสำเร็จรูป ผู้ใช้งานคอมพิวเตอร์จะต้องเรียนรู้อย่างน้อยที่สุดภาษาใดภาษาหนึ่ง เพื่อเขียนโปรแกรมสั่งงานที่เครื่องคอมพิวเตอร์จะเข้าใจได้ผู้ใช้จะต้องเขียนโปรแกรมให้ถูกต้องทั้งหลักไวยากรณ์และหลักการทำงานด้วยเหตุผล

ถ้าคอมพิวเตอร์มีความสามารถที่จะทำความเข้าใจภาษาพูด หรือ ภาษาเขียนของคนทั่วไปไม่ว่าจะเป็นภาษาไทย หรือ ภาษาอังกฤษแล้ว เราจะเห็นประโยชน์และคุณค่าอย่างมาก กล่าวคือถ้าเราสามารถทำให้คอมพิวเตอร์เรียนรู้ภาษาของมนุษย์ แทนที่มนุษย์จะต้องมาเรียนภาษาคอมพิวเตอร์ เพราะเมื่อคอมพิวเตอร์เข้าใจภาษาธรรมชาติได้แล้วจะทำให้บุคคลทั่วไปเพียงแค่อ่านออกเขียนได้ ในภาษาธรรมชาติภาษาใดภาษาหนึ่งเท่านั้นก็จะสามารถใช้งานคอมพิวเตอร์ได้อย่างมีประสิทธิภาพ

ยุคของการประมวลข้อมูลข่าวสาร

การประมวลผลทางด้านคอมพิวเตอร์นั้นนับตั้งแต่ยุคที่เริ่มมีคอมพิวเตอร์ใช้ (ค.ศ.1950) จนถึงปัจจุบัน แบ่งออกได้ 3 ยุค ดังนี้ (Marry Harris, 1985)

1. ยุคของการประมวลผลข้อมูล (Data Processing Age) เป็นยุคที่นำคอมพิวเตอร์มาคำนวณงานทางด้านวิทยาศาสตร์ ภาษาคอมพิวเตอร์ที่ใช้เช่น ภาษาฟอร์แทรน ภาษาโคบอล ฯลฯ การคำนวณจะกระทำเป็นงาน ๆ การจัดการทางด้านข้อมูลส่วนใหญ่จึงเน้นในระบบแฟ้ม (Filing System) คือนำข้อมูลจำนวนมากเก็บขึ้นเทปเมื่อต้องการใช้ก็นำข้อมูลจากเทปมาประมวลผล ระบบการประมวลผลแบบนี้จึงเรียกว่าระบบการประมวลผลแบบกลุ่ม (Batch Processing)

2. ยุคของข้อมูลข่าวสาร (Information Age) เป็นยุคที่ใช้ข้อมูลเป็นระบบมากขึ้น การประมวลผลจะพิจารณา เป็นระบบโดยนำข้อมูลหลาย ๆ แฟ้มมาสัมพันธ์เกี่ยวข้องกัน และเก็บรวบรวมไว้ในตัวกลางที่จะเรียกใช้ได้ตลอดเวลา ในยุคนี้จึงเกิดคำว่าฐานข้อมูล (Database) ในปัจจุบันการทำงานของคอมพิวเตอร์ได้ให้ความสำคัญทั่วไป เช่น ระบบออน์ไลน์(Online)ของธนาคารที่ต้องมีฐานข้อมูลกลาง ระบบการให้บริการข้อมูลข่าวสารจากฐานข้อมูลเฉพาะ เป็นต้น ปัจจุบันการออกแบบระบบมักเน้นในระบบฐานข้อมูล ดังนั้นเราจึงมีตัวช่วยจัดการฐานข้อมูลซึ่งเป็นซอฟต์แวร์ออกมาเป็นจำนวนมาก มาตรฐานของการจัดการฐานข้อมูลก็มีหลายรูปแบบซึ่งเหมาะกับระบบงานที่แตกต่างกันออกไป

3. ยุคของการประมวลความรอบรู้ (Knowledge Processing Age) เป็นยุคของคอมพิวเตอร์ที่เรากำลังเผชิญอยู่ขณะนี้ การตื่นตัวทางเทคโนโลยีเกี่ยวกับปัญญาประดิษฐ์ (AI) ทำให้เรากำลังหาหนทางนำเอาความรอบรู้ มาใช้ประโยชน์ได้มากขึ้น ระบบของซอฟต์แวร์ที่กำลังได้รับการกล่าวขวัญในยุคนี้คือ ระบบผู้เชี่ยวชาญ (Expert System) ระบบฐานความรู้ (Knowledge Base) และระบบช่วยให้คำปรึกษาและตัดสินใจต่าง ๆ การประมวลผลในยุคนี้จึงเกี่ยวข้องกับการใช้ภาษาธรรมชาติมากยิ่งขึ้น ระบบการใช้ฐานความรู้นี้เป็นศาสตร์ที่นำเอาเทคนิคการแทนความรอบรู้มาใช้ เมื่อมาประสานเข้ากัน ระบบการแทนความรู้ทางภาษาธรรมชาติจึงทำให้คอมพิวเตอร์เข้าใจภาษามนุษย์ได้ดีขึ้น

ระบบภาษาธรรมชาติ

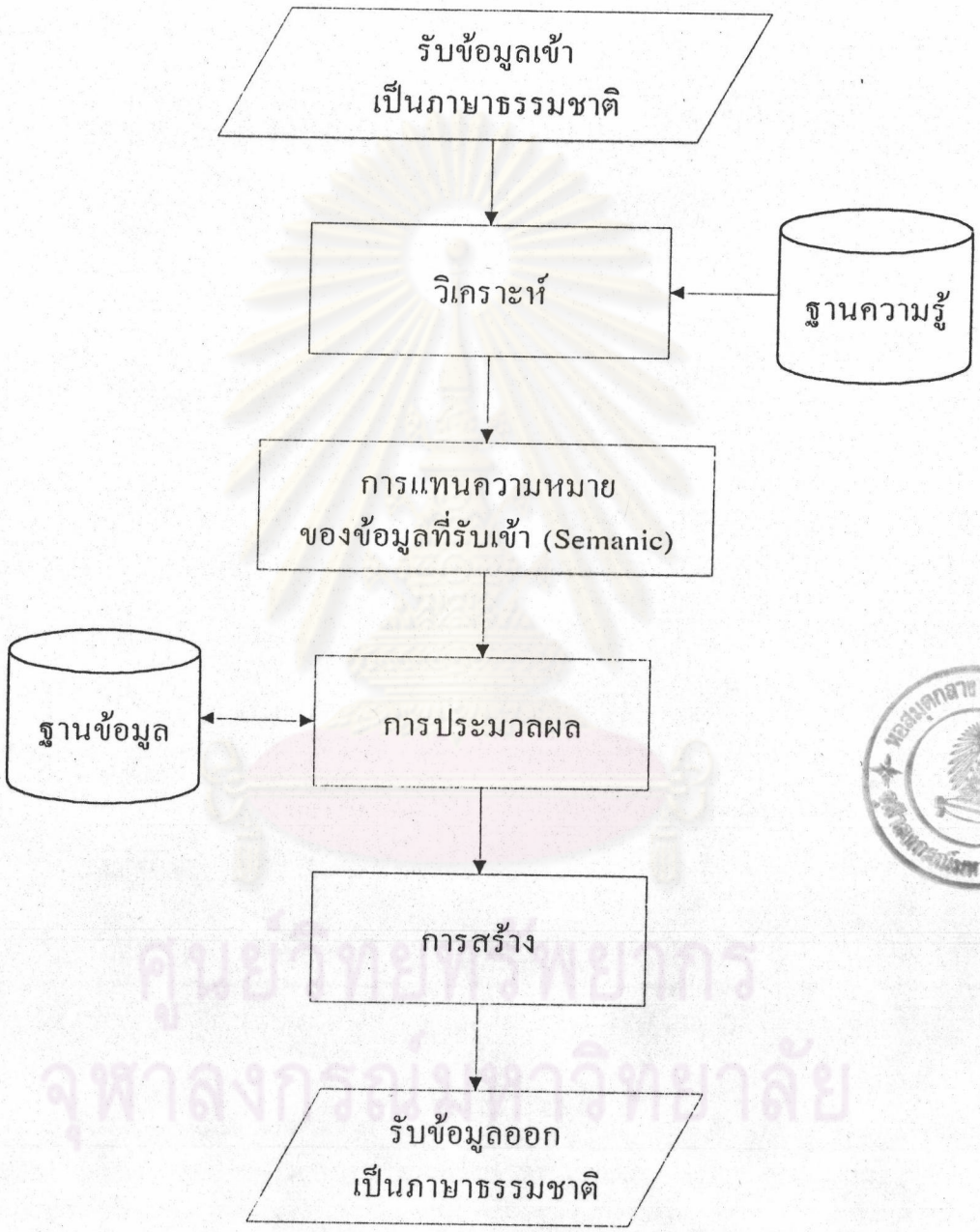
ระบบภาษาธรรมชาติเป็นระบบซอฟต์แวร์ที่กำลังได้รับการกล่าวถึงกันอย่างกว้างขวาง โดยจัดเป็นศาสตร์สาขาหนึ่งของปัญญาประดิษฐ์ซึ่งมีลักษณะและสมบัติที่สำคัญดังนี้ (Firebaugh,1989)

1. ส่วนของระบบรับข้อมูลเข้า รับข้อมูลออกที่ใช้ในการสั่งหรือติดต่อกับคอมพิวเตอร์จะอยู่ในลักษณะที่เป็นภาษาธรรมชาติ

2. การประมวลผลภายในระบบจะใช้หลักการพื้นฐานของฐานความเข้าใจของ ภาษาธรรมชาติ

3. เป็นกระบวนการติดต่อระหว่างผู้ใช้กับระบบเดิม เป็นการเข้าถึงหรือการเรียกดูจากฐานข้อมูล ซึ่งในลักษณะนี้จำเป็นต้องการวิเคราะห์ความหมายของประโยคเพื่อส่งไปให้หน่วยปฏิบัติดำเนินการต่อไป

ตัวแบบ (Model) ระบบภาษาธรรมชาติ จะมีลักษณะดังนี้



รูปที่ 3 ตัวแบบระบบภาษาธรรมชาติ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

เป้าหมายของเทคโนโลยีทางด้านการประมวลผลภาษาธรรมชาติในอนาคตที่สำคัญได้แก่ การรับรู้ด้วยเสียง (Speech Recognition)ซึ่งเป็นขบวนการรับรู้เสียงพูดนำไปวิเคราะห์ เพื่อแปลความหมายแล้วนำไปปฏิบัติ ส่วนอีกแขนงหนึ่งเป็นการใช้เครื่องคอมพิวเตอร์ทำหน้าที่แปลความหมายจากภาษาหนึ่งไปยังอีกภาษาหนึ่ง เช่น การใช้ภาษาธรรมชาติ ติดต่อกับภาษาสอบถามเชิงโครงสร้างเพื่อเข้าถึง(Access) ฐานข้อมูลที่มีอยู่แล้ว ซึ่งเป็นส่วนที่สนใจและนำมาศึกษาในการทำวิทยานิพนธ์นี้

การประมวลผลภาษาธรรมชาติในระดับต่าง ๆ

การประมวลผลภาษาธรรมชาติ แบ่งเป็นระดับต่าง ๆ ได้ดังต่อไปนี้ (Luger,1989)

1. ระดับฉันทลักษณ์ (Prosody) เป็นการศึกษาเกี่ยวกับฉันทลักษณ์ระดับสูงต่ำ และจังหวะของภาษาซึ่งขั้นตอนนี้ค่อนข้างยากที่จะจัดการ และไม่ค่อยมีความสำคัญในระบบคอมพิวเตอร์และมีใช้มากทางด้านบทกวี

2. ระดับเสียง (Phonology) เป็นการวิเคราะห์เสียงต่าง ๆ ที่ประกอบขึ้นมาเป็นภาษา ซึ่งเป็นงานเกี่ยวกับการรับรู้และสร้างเสียงพูด

3. ระดับหน่วยคำ (Morphology) เป็นการศึกษาเกี่ยวกับส่วนประกอบต่าง ๆ ที่ใช้สร้างคำ การใช้คำประกอบหน้า-หลัง (Prefix-Suffix) เพื่อขยายคำ การรวมคำ ผันคำ และความสัมพันธ์ของลูกคำ แม่คำ หรือการแตกกระจายของคำย่อย รวมทั้งการแบ่งแยกคำในประโยค

4. ระดับไวยากรณ์ (Syntactic) เป็นการวิเคราะห์ในเชิงโครงสร้าง ซึ่งคำต่าง ๆ ในภาษาธรรมชาติถูกนำมาเรียงกันไว้ และหลังจากให้ความหมายที่สัมพันธ์กันในที่สุดจะให้โครงสร้างของภาษาที่ชัดเจนออกมา แต่ในกรณีที่ประโยคที่อ่านเข้ามา ผิดหลักไวยากรณ์แล้วคอมพิวเตอร์ควรจะบอกได้ว่าเป็นประโยคที่ผิด

5. ระดับความหมาย (Semantic) เป็นการวิเคราะห์ความหมาย กล่าวคือประโยคที่ถูกต้องตามหลักทางไวยากรณ์จะต้องมีความหมายอย่างใดอย่างหนึ่งแน่นอน คอมพิวเตอร์ควรจะมีความสามารถในการหาความหมายของประโยคได้อย่างถูกต้อง เนื่องจากในประโยคหนึ่ง ๆ อาจจะให้ความหมายที่กำกวม หรือบางประโยคอาจจะมีคำศัพท์หรือรูปแบบโครงสร้างไวยากรณ์ที่ถูกต้องก็จริง แต่อาจจะไม่ให้ความหมายอะไรเลยก็ได้ คอมพิวเตอร์ควรจะมีความสามารถในการแยกความถูกต้องของความหมายได้ด้วย

6. ระดับการใช้ภาษา (Pragmatic) เป็นการวิเคราะห์เชิงปฏิบัติซึ่งหลังจากที่คอมพิวเตอร์เข้าใจโครงสร้างและความหมายของประโยคแล้ว ขั้นตอนที่สำคัญต่อมาก็คือการตีความ และลงมือกระทำตามความหมายหรือความเข้าใจนั้นต่อไป

7. ระดับความรู้ (Knowledge) เป็นการรวบรวมความรู้ต่าง ๆ ทั้งทางด้านวัตถุ และทางสังคมของมนุษย์ โดยมีจุดมุ่งหมายเพื่อใช้เป็นเครื่องมือติดต่อกับมนุษย์ในทุก ๆ เรื่องซึ่งจำเป็นต้องอาศัยความรู้ทางด้านผู้เชี่ยวชาญในสาขาต่าง ๆ เป็นจำนวนมาก เพื่อให้เครื่องคอมพิวเตอร์สามารถเข้าใจความหมายทั้งหมดของข้อความต่าง ๆ ในระบบการประมวลผลภาษาธรรมชาติ

การแจงประโยค (Parsing)

เป็นการวิเคราะห์โครงสร้าง กฎเกณฑ์ทางภาษา ไวยากรณ์ต่าง ๆ และความสัมพันธ์กันในเชิงภาษาอีกด้วย ซึ่งสามารถอธิบายด้วยแผนภาพต่าง ๆ ได้เช่น รูปแบบของประโยค ได้แก่ ประธาน-กริยา กริยา-กรรม นาม-ส่วนขยาย เป็นต้น ตามปกติจะใช้แผนภาพต้นไม้ในการอธิบายรูปแบบดังกล่าว และจะเรียกว่าการวิเคราะห์กระจายแบบต้นไม้ (Parse Tree) ซึ่งในขั้นตอนนี้เป็นการอาศัยความรู้เกี่ยวกับเรื่องไวยากรณ์ หน่วยคำ และระดับความหมาย เช่น

Somchai likes Somsri.

คอมพิวเตอร์จะต้องทำการวิเคราะห์แยกได้ว่า Somchai เป็นนาม กระทำกับกริยา likes และกรรม Somsri ซึ่งเป็นนามเช่นกัน

การวิเคราะห์ความหมายและความสัมพันธ์รูปแบบของความคิด โดยรวมเกี่ยวกับขอบเขต ความสัมพันธ์ทางตรรก และโครงสร้างทางภาษา รวมทั้งใช้ฐานความรู้ในเรื่องของ "คำ" มาช่วยอีกด้วย เช่นในตัวอย่างดังกล่าว

Somchai และ Somsri เป็นคน(นาม) กระทำกับกริยา likes โดยที่ Somchai เป็นผู้ชอบ Somsri

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

การค้นหาความหมายของประโยคจากฐานความรู้ที่เก็บไว้ในฐานความรู้การแปลความหมาย (Knowledge interpretation)เพื่อค้นหาเทียบเคียงกับที่เก็บไว้ เป็นการขยายความสัมพันธ์จากกฎเกณฑ์ทางภาษาไปสู่ความหมายในเชิงความรู้ที่เครื่องคอมพิวเตอร์สามารถเข้าใจได้ เช่นตัวอย่างที่กล่าวมาแล้วจะได้ความหมายว่า Somchai ชอบ Somsri

เทคนิคต่าง ๆ ในการประมวลผลภาษาธรรมชาติ

เนื่องจากขอบเขตของการประมวลผลทางภาษาด้วยคอมพิวเตอร์ค่อนข้างกว้างขวางจึงแทบจะเป็นไปไม่ได้ที่จะให้การประมวลผลครอบคลุมในทุก ๆ กรณี ดังนั้นจึงต้องมีการกำจัดความซับซ้อน และความยุ่งยากทางภาษาต่าง ๆ ให้อยู่ในขอบเขต โดยเฉพาะการประมวลผลภาษาธรรมชาติหรือภาษาของมนุษย์ที่มีความยืดหยุ่นและความซับซ้อนมาก เมื่อเป็นเช่นนี้ จึงต้องหาวิธีทำให้หน่วยประมวลผล สามารถเข้าใจรูปแบบของประโยค เป็นรูปแบบมาตรฐานทั่ว ๆ ไป และเป็นประโยคบอกเล่าซึ่งประกอบด้วย ไปด้วย

ประธาน กริยา กรรม
Subject, Verb, Object

สมมุติว่าในเครื่องคอมพิวเตอร์ มีการเก็บคำ (Word) และชนิดของคำ (Type) ที่จะใช้อธิบายในบทนี้ดังนี้

Word	Type
door	noun
window	noun
house	noun
child	noun
has	verb
runs	verb
plays	verb
large	adjective

Word	Type
quickly	adverb
the	determiner
a	determiner
to	preposition

ตารางที่ 11 แสดงคำและชนิดของคำ

นอกจากนี้ประโยคก็อาจจะมีคำขยาย คำนาม หรือ คุณศัพท์ (Adjective) และคำขยายกริยา หรือ กริยาวิเศษณ์ (Adverb) และจบด้วยเครื่องหมาย "." มหัพภาค (Period) เช่นประโยคต่อไปนี้

The child runs to the house.

The large child runs quickly to the window.

สมมุติว่าเราเลือกแจกแจงประโยค

The child quickly runs to the house.

จะเห็นว่าไม่ถูกต้องเพราะคำกริยาวิเศษณ์ "quickly" ขยายคำกริยา "runs" ไม่ถูกกฎเกณฑ์ที่สร้างเก็บไว้ในคอมพิวเตอร์ ยิ่งไปกว่านั้น ยังจะต้องมีการเก็บคำศัพท์ (Vocabulary) และชนิดของคำศัพท์ที่จำเป็นไว้ด้วย คำศัพท์นี้ต้องสามารถป้อนเพิ่มเติมได้อีกเท่าที่ต้องการ

พื้นฐานของระบบการประมวลผลภาษาธรรมชาติ ก็คือ วิธีการวิเคราะห์แบบกระจาย (Parser) ซึ่งเป็นเครื่องมือตรวจสอบส่วนของรหัสที่อ่านเข้ามาทีละคำในแต่ละประโยค เพื่อที่จะได้ทราบว่าแต่ละคำนั้นเป็นคำชนิดใด วิธีการวิเคราะห์แบบกระจายที่ได้ศึกษามี 3 วิธีดังนี้

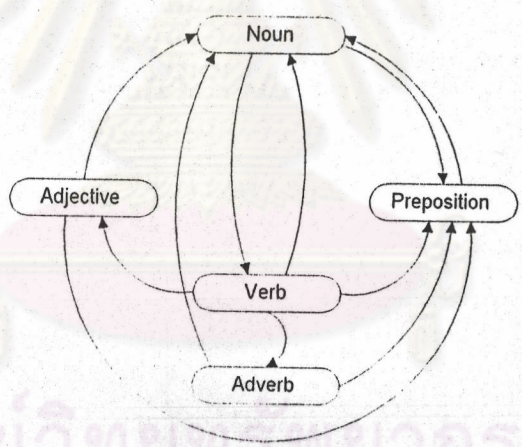
1. วิธีการวิเคราะห์แบบกระจายโดยการใช้กราฟบอกทิศทางจากสถานะหนึ่งไปอีกสถานะหนึ่ง (State-Machine Parser)

2. วิธีการวิเคราะห์แบบกระจายที่ใช้หลักไวยากรณ์ไม่พึงบริบท (Context - Free Grammar Parser)

3. วิธีการวิเคราะห์แบบกระจายโดยการกำจัดคำออกไป (Noise-Disposal Parser)

วิธีการวิเคราะห์แบบกระจายโดยการใช้กราฟบอกทิศทางจากสถานะหนึ่งไปอีกสถานะหนึ่ง (State-Machine Parser)

วิธีการวิเคราะห์แบบกระจายเป็นการใช้สภาวะปัจจุบัน หรือคำที่กำลังกล่าวถึง เป็นตัวทำนายคำต่อไปว่าจะต้องเป็นคำชนิดใด ซึ่งสามารถใช้กราฟบอกทิศทาง (Directed Graph) แสดงความสัมพันธ์จากสถานะหนึ่งไปยังอีกสถานะหนึ่ง ดังรูปจะเห็นว่าคำนามสามารถมีกริยาหรือคำขยายนาม ตามมาข้างหลังได้เป็นต้น



รูปที่ 4 วิธีการวิเคราะห์แบบกระจาย

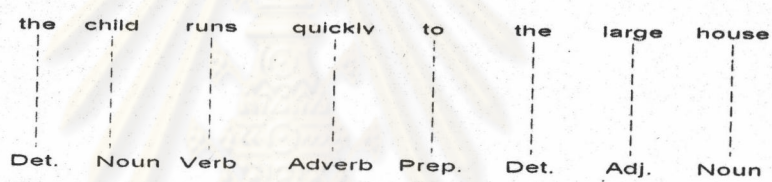
จากกราฟดังกล่าว จะสามารถบอกได้ว่า จากสถานะหนึ่งไปยังคำต่อไปสามารถเป็นสถานะไหน ได้บ้าง และเราสามารถตรวจสอบ ประโยคที่ป้อนเข้ามาว่ามีไวยากรณ์ตรงกับที่ต้องการหรือไม่ โดยไม่จำเป็นต้องบังคับรูปแบบประโยคจนเกินไป เราสามารถแทนที่กราฟนี้ด้วยโปรแกรม ซึ่งจะทำการตรวจสอบคำทีละคำ ว่าเป็นชนิดใด เมื่อทราบคำแรก ก็สามารถตรวจคำต่อไปว่าสามารถเป็นคำอะไรได้บ้าง เมื่อตรวจสอบจนกระทั่งจบประโยคแล้ว ถ้าไม่พบความผิดพลาดใด ๆ แสดงว่าประโยคนี้ถูกต้องตามไวยากรณ์ที่เราได้กำหนดไว้ในกราฟนั่นเองดังนั้นจะต้อง

มีการอธิบายไวยากรณ์ให้อยู่ในรูปของกราฟ ซึ่งเราสามารถประยุกต์เข้ากับภาษาใด ๆ ก็ได้ โดยทำการวิเคราะห์ภาษานั้นโดยละเอียด แล้วกำหนดกฎเกณฑ์ที่ต้องการลงบนกราฟ นอกจากนี้ยังจะต้องเก็บคำต่าง ๆ ลงในฐานข้อมูลเพื่อให้ระบบรู้จัก พร้อมทั้งชนิดของคำนั้น ๆ ด้วย จากกราฟดังกล่าวสามารถเขียนเป็นโปรแกรมย่อยโดยใช้ภาษาซี ได้ดังนี้

```
do{
  word = get_token();
  type = get_type(word);
  if (type==DBT) turn stste;
  if (type==TERM) return TERM;
        switch(stste){
          case STARTUP;
                if(type! = DET)
                  return type;
                  else
                        return STARTUP;
         case NOUN;
                if (type==VERB) return VERB;
                if (type==PREP) return PREP;
                break;
        case VERB;
                if (type==PREP) return PREP;
                if (type==NOUN) return NOUN;
                if (type==ADV) return ADV;
                if (type==ADJ) return ADJ;
                break;
        case ADV;
                if (type==NOUN) return NOUN;
                if (type==PREP) return PREP;
                break;
```

```
                case ADJ;
                        if (type==NOUN) return NOUN;
                        break;
                case PREP;
                        if (type==ADJ) return ADJ;
                        if (type==NOUN) return NOUN;
                        break;
                }
        } while word != '.';
```

จากโปรแกรมย่อยดังกล่าวเมื่อนำมาวิเคราะห์ประโยคได้ผลดังนี้



รูปที่ 5 การวิเคราะห์ประโยค

ระบบจะทำการอ่านคำแรก the ซึ่งมีรูปแบบเป็นคำนำหน้า จึงไม่มีการเปลี่ยนสถานะ คำต่อไป คือ Child มีรูปแบบเป็นคำนาม(Noun) ทำให้มีการเปลี่ยนสถานะเป็นคำนามคำต่อไป คือ runs มี รูปแบบเป็นคำกริยา(Verb) แต่เนื่องจากกราฟบอกว่าจากสถานะเป็นคำนาม สามารถเป็นได้ทั้งคำกริยา และคำบุพบท (Preposition) ดังนั้นคำนี้จึงถูกหลักไวยากรณ์ และเปลี่ยนสถานะ เป็นคำกริยา

คำต่อไปคือ quickly มีรูปแบบเป็นคำกริยาวิเศษณ์ (Adverb) แต่จากสภาวะก่อนเป็นคำกริยาซึ่งสามารถต่อท้ายด้วยคำกริยาวิเศษณ์ได้ ทำนองเดียวกันคำต่อไปคือ to ซึ่งเป็น คำบุพบท สามารถตามหลังคำกริยาวิเศษณ์และทั้งสามารถตามด้วย คำนำหน้า คำคุณศัพท์ และคำนามตามลำดับ

วิธีการวิเคราะห์แบบกระจายโดยใช้กราฟ State-Machine สามารถทำงานได้เป็นอย่างดี แต่มีข้อจำกัดความยุ่งยากของภาษา โดยเฉพาะกรณีที่ภาษามีความซับซ้อนมาก แม้ว่าจะเป็นภาษาอังกฤษก็ตาม ดังนั้นจึงต้องจำกัดไวยากรณ์ของภาษาให้อยู่ในขอบเขตที่กำหนด ตัวอย่าง เช่น ภาษาควบคุมการทำงาน (Job Control Language : JCL) ในระบบปฏิบัติการ หรือภาษาสั่งงานการสอบถามจากฐานข้อมูลเป็นต้น

วิธีการวิเคราะห์แบบกระจายโดยใช้หลักไวยากรณ์ไม่พึงบริบท (Context - Free Grammar Parser)

ในการกระจายคำในประโยคอีกวิธีหนึ่ง ใช้หลักการแบ่งคำในประโยคออกเป็นส่วนย่อย แล้วจึงแบ่งแต่ละส่วนย่อยซ้ำไปเรื่อย ๆ (Recursive Descent) จนกว่าจะได้คำย่อยที่สุด ที่ไม่สามารถแบ่งได้อีก (Atomic Word) และระบบจะรู้จักคำดังกล่าว ซึ่งได้แก่ คำนาม คำกริยา คำคุณศัพท์เป็นต้น จากนั้นก็จะรวบรวมความหมายของแต่ละคำย่อยมา ประกอบกันเป็นความหมายที่สมบูรณ์มากขึ้น และได้ความหมายของประโยคที่สมบูรณ์ที่สุด โดยอาศัยกฎการผลิต (Production Rule) ซึ่งจะต้องมีการป้อนกฎเกณฑ์การสร้างคำ โครงสร้างต่าง ๆ และความหมายไว้ล่วงหน้า ดังตัวอย่างต่อไปนี้

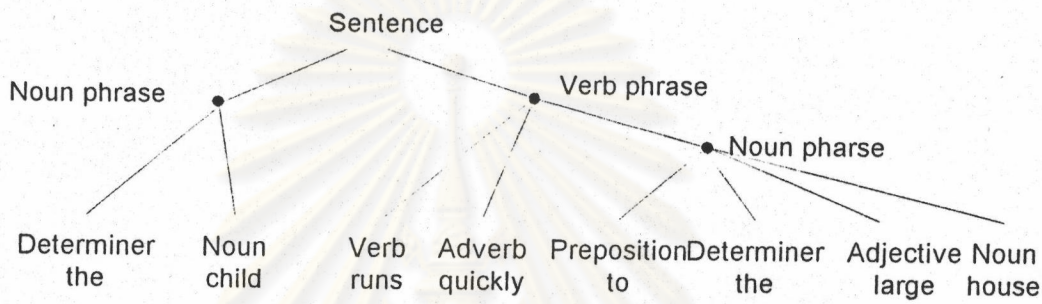
SENTENCE : = NP + VP
NP : = determiner + noun
NP : = determiner + adjective + noun
NP : = preposition + NP
VP : = verb + NP
VP : = verb + adverb + NP
VP : = verb + adverb
VP : = verb

รูปที่ 6 แสดงการใช้กฎการผลิต

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

จากกฎเกณฑ์การผลิตข้างต้น จะเห็นได้ว่ามีกระบวนการการกระทำแบบเรียกใช้ตนเอง (Recursive) เช่น นามวลี (NP) จะมีการกระทำซ้ำเมื่อมีคำบุพบทนำหน้า หรือกริยาวลี (VP) ก็มีการเลือกใช้นามวลีเมื่อมีคำกริยานำหน้า เป็นต้น เมื่อลองใช้กฎดังกล่าว กับประโยคตัวอย่างจะได้ผลดังนี้

The child runs quickly to the large house.



รูปที่ 7 การกระจายแบบต้นไม้ (Parse tree)

จากรูปข้างบนจะเห็นได้ว่า สามารถแยกประโยคดังตัวอย่างให้อยู่ในรูปของต้นไม้ (Tree) ซึ่งเรียกว่าการกระจายแบบต้นไม้ (Parse tree) โดยที่ยังไม่สนใจในความหมายของแต่ละคำ เพียงแต่สนใจไวยากรณ์ เท่านั้น ซึ่งวิธีนี้มีชื่อเรียกว่าการไม่พึงบริบท (Context free)

จากแนวคิดดังกล่าว ได้มีการประยุกต์ใช้ในภาษาคอมพิวเตอร์แทบทุกภาษา เช่น ภาษาปาสคาล ภาษาเบสิก ภาษาซี และอื่น ๆ นอกจากนี้ยังสามารถนำไปประยุกต์ใช้กับภาษาอังกฤษซึ่งกระทำได้โดย การสร้างกฎเกณฑ์ของภาษาขึ้น เพื่อใช้ในการสร้างวลีขึ้นจากคำ สร้างประโยคขึ้นจากคำและวลี ในทางตรงกันข้าม ก็สามารถแยกประโยคเป็นวลี แยกวลีเป็นคำได้เช่นเดียวกัน

ส่วนการเขียนโปรแกรมเพื่อใช้เทคนิควิธีนี้ ก็สามารถทำได้ แต่เนื่องจากคุณสมบัติของภาษารุ่นใหม่ ๆ สามารถใช้เทคนิคการเขียนโปรแกรมแบบการเรียกซ้ำ (Recursive) ได้โดยใส่กฎเกณฑ์ ทางไวยากรณ์ลงไป ดังตัวอย่างต่อไปนี้

```
/* context-free grammar parser */
parse()
{
   if (!nounphrase()) return 0;
   if (!verbphrase()) return 0;
   if (!terminator()) return 0;
   return 1;
}
/* read a noun phrase from the input stream */
nounphrase()
{
  char type();
  get_token();
  type=find_type(token);
  switch(type); {
     case DET:
        get_token();
        type=find_type(token);
        if (type==NOUN) return 1;
        else if (type==ADJ)  {
        get_token();
        type=find_type(token);
        if (type==NOUN) return 1;
     }
    break;
 case PREP:
    return nounphrase();
 }
 return 0;
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
}
/* read a verb phrase */
verbphrase ()
{
  char type,*pos;
  get_token ();
  type=find_type(token);
  if(type!=VERB) return 0;  /* must start with a verb */
  pos=t_pos;  /* save current position for backtracking */
  /* verb + adverb + NP */
  if(verb_adv_np()) return 1;
  /* verb + NP */
  t_pos=pos;  /* back up */
  if (verb_np()) return 1;
  /* verb + adverb -- no NP */
  t_pos=pos;
  if (verb_adv()) return 1;
  /* just verb */
  return 1;
}
verb_np()
{
  /* verb + NP */
  return nounphrase();
}
verb_adv_np()
{
  char type();
  get_token();
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
    type=find_type(token);
    if (type==ADV && nounphrase()) return 1;
    return 0;
}
verb_adv()
{
    char type;
    get_token();
    type=find_type(token);
    return (type==ADV);
}
terminator()
{
    get_token();
    return (find_type(token) ==TERM);
}
```

การทำงานของโปรแกรมเริ่มต้นจาก ฟังก์ชันวลี (Parse()) ซึ่งจะทำการตรวจสอบว่าเป็นฟังก์ชัน กริยาวลี (Verb()) ,นามวลี(Noun()), Terminator ( ) หรือไม่ ถ้าใช่จะส่งค่ากลับไปเป็นจริง มิฉะนั้นจะส่งค่าไม่จริงกลับไป และจะรายงานความผิดพลาด

ในทำนองเดียวกันกริยาวลีก็จะทำการตรวจสอบว่าเป็น Verb + Adverb + NP,Verb + NP,Verb + Adverb,Verb หรือไม่ ซึ่งก็คือกฎเกณฑ์ที่ได้กำหนดไว้แล้วนั่นเอง จากนั้นก็จะมีการตรวจสอบในลำดับถัดไป จนกว่าจะได้ว่าเป็น Noun, Verb, Adjective, Terminator และได้เก็บค่าไว้ในฐานความรู้ล่วงหน้านั้นแล้ว สำหรับในส่วนของการค้นหารูปแบบของคำ (Find_Type( )) จะทำหน้าที่ตรวจสอบรูปแบบของคำนั้น ๆ

ข้อได้เปรียบของวิธีนี้ คือ ง่ายต่อการจัดทำโปรแกรมด้วยภาษาระดับสูง สามารถจัดการแยกประเภทเป็นคำ หรือ วลีได้ นอกจากนี้ยังสามารถตรวจสอบตำแหน่งที่อยู่ของคำต่าง ๆ ในประโยคได้ง่าย ส่วนข้อเสียก็คือ ความซับซ้อนของภาษา อาจก่อให้เกิดการเรียก ซ้ำ ๆ กันจนมากเกินไป

วิธีการวิเคราะห์แบบกระจายโดยการกำจัดคำออกไป (Noise - Disposal Parser) (Schildt, 1987)

เป็นเทคนิคอีกอย่างหนึ่ง ซึ่งนิยมใช้การประมวลผลหลาย ๆ ด้าน โดยอาศัยหลักการกำจัดคำส่วนที่ไม่ต้องการออกไป คงเหลือไว้แต่คำหลัก (Keyword) เท่านั้น ทั้งนี้เพราะว่าประโยคภาษาธรรมชาติ จะมีส่วนที่เป็นคำที่ไม่ได้มีความหมายหลัก เช่น คำทักทาย คำแสดงอารมณ์ ส่วนขยายของคำ เป็นต้น สำหรับกรณีการประมวลผลภาษาธรรมชาติ จะมีการกำหนดคำที่ระบบรับรู้รวมทั้งความหมายของคำนั้นไว้ล่วงหน้า คำใดที่ไม่มีในฐานความรู้จะถือเป็นส่วนเกิน (Noise) และถูกกำจัดออกไปโดยมีไวยากรณ์ที่กำหนดไว้สำหรับสั่งงานระบบ ซึ่งจะมีรูปแบบไวยากรณ์ดังนี้

COMMAND < MODIFIER > < NAME > < OPERATOR > < VALUE >

รูปที่ 8 รูปแบบไวยากรณ์แบบการกำจัดคำออกไป

จากรูปแบบในรูปที่ ส่วนที่อยู่ในเครื่องหมาย <.....> หมายถึงมีก็ได้ ไม่มีก็ได้และประโยคจะต้องประกอบด้วยคำสั่ง (Command) เสมอ อาจมีส่วนขยาย(Modifier) หรือชื่อ เอ็นทิตี้ (Entity) ใด ๆ หรือมีการกระทำทางคณิตศาสตร์ (Operator) เปรียบเทียบกับค่าใด ๆ เช่น

Show me all companies with stock prices > 100

Show me all

is stock selling = 40?

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

โดยที่ Show, is เป็น Command

me, all เป็นส่วนขยาย

stock, Companies เป็น Name.

>, = เป็น Operator

ซึ่งสามารถเขียนเป็นโปรแกรม เพื่อจัดการตรวจสอบคำได้ดังนี้

```
/* get the command */
get_com(s)
char *s;
{
  get_noise();
  get_token();
  if (find_type(token)!=COMMAND) {
    strcpy (s,"");
    return 0;
  }
  strcpy (s,token);
  return 1;
}
/* get a modifier or default to "all" */
get_mod (s)
char *s;
{
  get_noise();
  get_token();
  if (find_type(token) ! = MODIFIER) {
    strcpy (s,"all");
    put_back ();
    return;
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
    }
    strcpy (s,token);
  }
  /* get the command */
  get_name (s)
  char *s;
  {
    get_noise();
    get_token();
    if (find_type(token) ! = NAME)  {
      strcpy (s,"");
      put_back ();
      return;
    }
    strcpy (s,token);
  }
  /* get an operator */
  get_op()
  {
    get_noise();
    get_token();
    if (find_type(token) == OPERATOR) {
      return *token;
    else return 0;
  }
  /* get a number */
    float t;
    get_token();
    if (*token!='.')
      sscanf (token,"%f",&t);
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
    return t;
}
/* check for noise */
get_noise()
{
   do {
      get_token();
   } white (!find_type (token));
   put_black();  /* return a valid token to stream */
}
```

โปรแกรมนี้ได้เขียนขึ้นโดยอาศัยสมมุติฐานดังกล่าวมาแล้ว โดยมีคำหลักที่ถูกบรรจุในฐานความรู้คำที่ไม่ใช่คำหลักก็จะถูกกำจัดออกไป และจะไม่สนใจในความหมาย จากโปรแกรมตัวอย่างจะให้ผลลัพธ์ดังนี้

```
Show me all Companies with price > 100
UCK
Show me all
UCL
ABC
XYZ
Please show me one with a price < 50
ABC 35
Is the price UCL > 100 ?
Yes
```

จะเห็นได้ว่า ในตัวอย่างดังกล่าว ประกอบด้วยคำสั่งเพียง 2 คำสั่ง เท่านั้น คือ

Show เพื่อแสดงข้อมูลที่มีอยู่

Is เพื่อสอบถามว่ามีอยู่หรือไม่

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

นอกจากนี้ เรายังสามารถเพิ่มคำสั่งเข้าไปในฐานความรู้ได้เท่าที่ต้องการ ทำให้เราสามารถสอบถามด้วยคำสั่งดังกล่าวเกี่ยวกับฐานข้อมูลได้

ข้อเสียของวิธีการนี้ก็คือ ประโยคสอบถามจะต้องมีลักษณะตามกฎเกณฑ์ที่กำหนดไว้ นั่นคือ ประกอบด้วย คำสั่ง ส่วนขยาย ชื่อ การกระทำทางคณิตศาสตร์และค่าต่าง ๆ เป็นต้น

ความยุ่งยากในการประมวลผลภาษาธรรมชาติ

แม้ว่าการประมวลผลภาษาธรรมชาติในปัจจุบัน จะมีเทคนิคและรูปแบบไวยากรณ์ใหม่ ๆ เกิดขึ้นอีกเป็นจำนวนมาก มีระบบวิเคราะห์กระจายที่ประยุกต์ใช้อย่างแพร่หลาย ในงานวิเคราะห์ภาษาที่จัดอยู่ในกลุ่มของปัญญาประดิษฐ์หลายระบบ แต่การประมวลผลภาษาธรรมชาติก็ยังมีความยุ่งยากอยู่หลายประการดังนี้ (Rich,1983)

1. ความกำกวมของคำ บางภาษามีลักษณะพิเศษ เช่น คำคำเดียว อาจมีได้หลายความหมาย ขึ้นอยู่กับความหมายของประโยคโดยรวม หรือขึ้นอยู่กับคำที่อยู่ก่อนหน้าหรือตามหลัง เป็นต้น

2. ความหลากหลายในรูปแบบของภาษา ซึ่งเป็นลักษณะพิเศษของแต่ละภาษา เช่น ประโยคความหมายเดียวกัน สามารถเขียนได้ในหลายรูปแบบ

3. ความซับซ้อนของส่วนขยายคำ เช่นการอ้างถึง ของคำบุพบทที่มีคำนามหลายตัว

4. คำเฉพาะ ภาษาถิ่น ศัพท์แสลงต่าง ๆ ซึ่งใช้ในวงแคบ ๆ หรือใช้ในแต่ละวงการเท่านั้น

5. ความซับซ้อนของแต่ละภาษา ทำให้การแปลความหมายคลาดเคลื่อนไป เช่น บางภาษาต้องแปลความหมายจากต้นประโยคไปท้ายประโยค หรือบางภาษาแปลความหมายจากท้ายประโยคมาต้นประโยค เป็นต้น