บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

บทนี้จะกล่าวถึง ทฤษฎีและงานวิจัยที่เกี่ยวข้องในวิทยานิพนธ์ฉบับนี้

ส่วนของทฤษฎีที่นำมาใช้ในการทำวิจัย ประกอบไปด้วย ทฤษฎีในศาสตร์การค้นคืนข้อมูล ทฤษฎีในศาสตร์ปัญญาประดิษฐ์

ส่วนของงานวิจัยที่นำมาเป็นแนวทางการศึกษาเพื่อทำวิจัยนั้น ประกอบไปด้วย งานวิจัยและบทความที่มีการประยุกต์ใช้เทคนิคการตัดคำ งานวิจัยที่มีการประยุกต์ใช้เทคนิคการวิเคราะห์หาความหมายแอบแฝง และ งานวิจัยที่มีการประยุกต์ใช้เทคนิคโครงข่ายประสาทเทียม ดังจะแสดงรายละเอียดของทฤษฎี หลักการ และแนวทางการศึกษาเพื่อทำงานวิจัยชิ้นนี้ ดังต่อไป

2.1 ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีในศาสตร์การค้นคืนข้อมูลที่เกี่ยวข้องกับงานวิจัยชิ้นนี้ ประกอบไปด้วย แบบจำลองเวกเตอร์ สำหรับศาสตร์การค้นคืนข้อมูล (Vector Model for Information Retrieval) การตัดคำ (Word Segmentation) น้ำหนักของคำ (Term Weight) พีชคณิตเชิงเส้นสำหรับศาสตร์การค้นคืนข้อมูล (Linear Algebra for Information Retrieval) และเทคนิคการวิเคราะห์หาความหมายแอบแฝง

ทฤษฎีในศาสตร์ปัญญาประดิษฐ์ที่เกี่ยวข้องกับงานวิจัยชิ้นนี้ ประกอบไปด้วย เทคนิคโครงข่ายประสาทเทียม อัลกอริทึมการแพร่กระจายย้อนกลับ และ เทคนิคโครงข่ายประสาทเทียมแบบการแพร่กระจายย้อนกลับ

2.1.1 ทฤษฎีในศาสตร์การค้นคืนข้อมูล

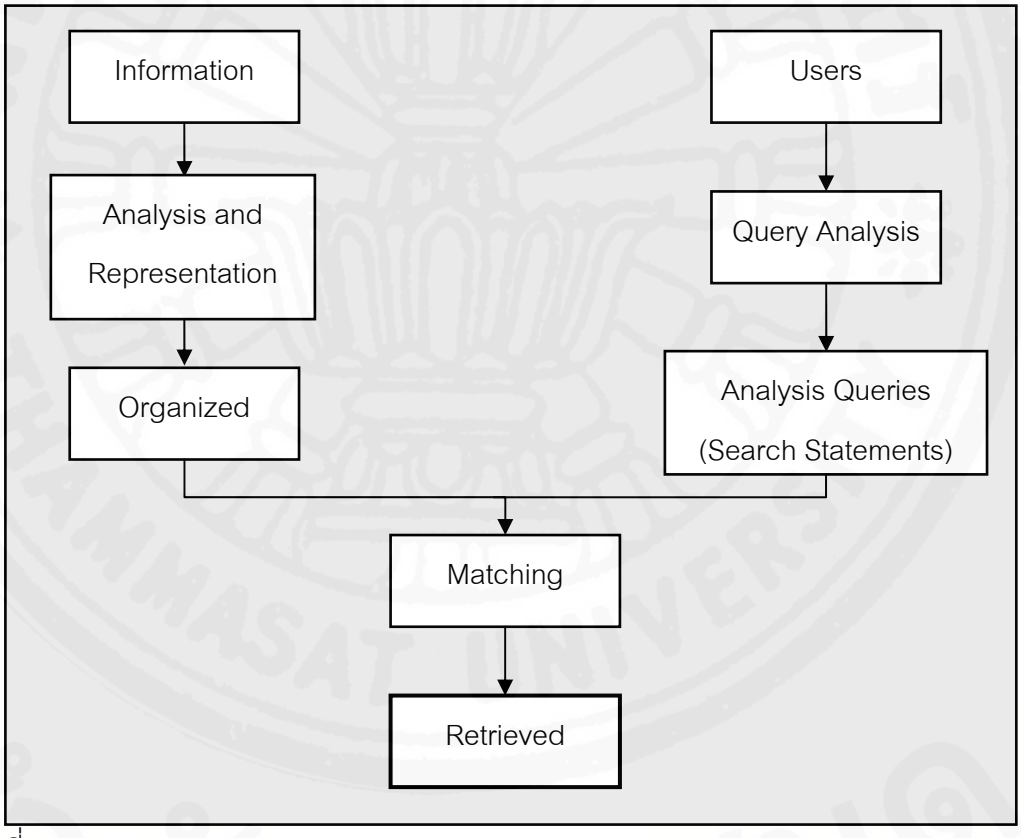
กระบวนการค้นหาความหมายแอบแฝง กล่าวได้ว่าเป็นกระบวนการหนึ่งในศาสตร์การค้นคืนข้อมูลที่ทำให้สามารถค้นคืนเอกสารได้ตรงประเด็นหรือตรงตามความต้องการ โดยที่กระบวนการดังกล่าว จะไม่ทำการค้นหาเอกสารจากการพิจารณาเพียงแค่รูปแบบของเอกสารที่เหมือนกัน แต่จะพิจารณาจากการวิเคราะห์หาความหมายแอบแฝงของเอกสารเพื่อเปรียบเทียบกับประเด็นที่ต้องการค้นหา จึงทำให้ระบบการค้นคืนข้อมูล (Information Retrieval System)

สร้างความพึงพอใจและความเชื่อมั่นให้กับผู้ใช้งานระบบในการค้นคืนเอกสารตรงตามความต้องการ (Grossman and Frieder, 2004)

ระบบการค้นคืนข้อมูลนำเสนอหลักการการทำงานที่สำคัญ คือ การสืบค้นและการค้นคืน โดยกระบวนการของการทำงาน ประกอบไปด้วย การวิเคราะห์ความต้องการของผู้ใช้งานระบบ การสร้างสูตรการค้นหา การค้นหา และการค้นคืนข้อมูล ดังแสดงแผนภาพการทำงานของระบบการค้นคืนข้อมูล ในภาพที่ 2.1

ภาพที่ 2.1

แผนภาพแสดงการทำงานของระบบการค้นคืนข้อมูล



ที่มา: Chowdhury (2004)

กลุ่มนักวิจัยและนักพัฒนาระบบเฉพาะทางในสาขาการค้นคืนข้อมูล ให้การรับรองว่าทฤษฎี หลักการและวิธีปฏิบัติในศาสตร์การค้นคืนข้อมูลนั้น สามารถนำไปประยุกต์ใช้เพื่อการออกแบบระบบที่มีความสามารถในการระบุองค์ประกอบสำคัญของเอกสาร สามารถวิเคราะห์เนื้อหาของเอกสาร และสามารถแสดงขั้นตอนการประมวลผลเอกสารได้ เป็นต้น ในอีกแง่มุมหนึ่ง

กลุ่มนักวิจัยได้ทำการศึกษาค้นคว้าเพื่อพัฒนาเทคนิคการค้นหาให้มีความสามารถในการเรียนรู้มากยิ่งขึ้น อีกทั้งพัฒนาเทคนิคที่หลากหลายสำหรับการให้ผลลัพธ์ที่เท่าเทียมกันทั้งผู้ใช้งานทั่วไปและผู้ใช้งานผ่านระบบเครือข่ายอีกด้วย (Chowdhury, 2004)

2.1.1.1 แบบจำลองเวกเตอร์สำหรับศาสตร์การค้นคืนข้อมูล

แบบจำลองเวกเตอร์ (Vector Model) เป็นแบบจำลองเชิงพีชคณิต (Algebraic Model) และได้มีการนำแบบจำลองเวกเตอร์มาประยุกต์ใช้ ใน ศาสตร์การค้นคืนข้อมูล ศาสตร์การกรองข้อมูล (Information Filtering) การทำดัชนีคำ (Indexing) และการจัดลำดับความสัมพันธ์ (Relevancy Ranking) ซึ่งจากการนำแบบจำลองเวกเตอร์มาประยุกต์ใช้ในหลากหลายแนวทางเช่นนี้ แสดงให้เห็นว่า แบบจำลองเวกเตอร์สามารถนำมาเป็นตัวแทนของเอกสารได้ โดยเอกสารที่เกี่ยวกับภาษาธรรมชาติจะสามารถใช้แบบจำลองเวกเตอร์หลายมิติในการเป็นตัวแทนเอกสาร ซึ่งผลการศึกษาวิจัยในเรื่องดังกล่าวเป็นผลงานวิจัยจาก มหาวิทยาลัยคอร์เนล (Cornell University) ที่ได้นำหลักการในศาสตร์การค้นคืนข้อมูลมา เพื่อพัฒนาในส่วนของงานวิจัยบนระบบ SMART ซึ่งเป็นระบบแรกที่นำหลักการของแบบจำลองเวกเตอร์สำหรับศาสตร์การค้นคืนข้อมูล มาประยุกต์ใช้อย่างได้ผล (“SMART Information Retrieval System”, Wikipedia Encyclopedia, 2006)

ในวิทยานิพนธ์ฉบับนี้นำเสนอ แบบจำลองเวกเตอร์ สำหรับศาสตร์การค้นคืนข้อมูล โดยจะกล่าวถึง คำนิยาม และตัวอย่างการนำไปประยุกต์ใช้ ตามลำดับดังต่อไปนี้

คำนิยามของแบบจำลองเวกเตอร์

สำหรับแบบจำลองเวกเตอร์นั้นประกอบไปด้วยสมาชิกที่เป็น น้ำหนักของคำ (The Weight, $w_{i,j}$) ซึ่งมีความสัมพันธ์กันกับ คู่อันดับของ ดัชนีของคำโดยทั่วไป (Generic Index Term, k_i) กับ เอกสาร (Document Term, d_j) โดยสามารถเขียนเป็นคู่อันดับได้ดังนี้ (k_i, d_j)

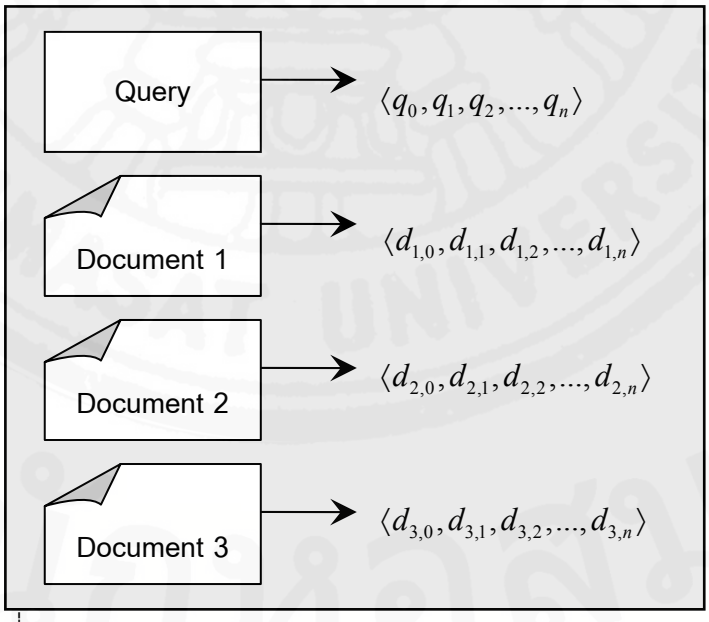
คำนิยามของแบบจำลองเวกเตอร์ มีการกำหนดให้ $w_{i,j}$ เป็นน้ำหนักของคำที่มีความสัมพันธ์กันกับคู่อันดับ $\left[k_i, d_j\right]$ โดยค่าของน้ำหนักของคำต้องมากกว่าหรือเท่ากับศูนย์ ($w_{i,j} \geq 0$)จากนั้นจึงนำน้ำหนักของคำมาสร้างเวกเตอร์ที่เป็นตัวแทนของคำที่ต้องการสืบค้น (Query Vector, $\vec{q}$) ได้ดังนี้ $\vec{q} = (w_{1,q}, w_{2,q}, ..., w_{t,q})$ โดย t คือผลรวมของดัชนีของคำทั้งหมด (Index Term) ที่ปรากฏในเอกสาร และสามารถสร้างเวกเตอร์ที่เป็นตัวแทนของเอกสาร

(Document Vector, d_j) ได้ดังนี้ $\overrightarrow{d_j} = (w_{1,j}, w_{2,j}, ..., w_{t,j})$ โดย t คือผลรวมของดัชนีของคำทั้งหมด (Index Term) ที่ปรากฏในเอกสาร เช่นเดียวกัน (Yates and Neto,1999)

การนำแบบจำลองเวกเตอร์ไปประยุกต์ใช้

โดยรูปแบบหนึ่งในการนำแบบจำลองเวกเตอร์ไปประยุกต์ใช้ นั่นก็คือ การวัดค่าความคล้ายคลึงกันระหว่างเอกสารกับคำที่ต้องการสืบค้น โดยกำหนดให้นำเวกเตอร์ตัวแทนแต่ละเอกสารและเวกเตอร์ตัวแทนของคำที่ต้องการสืบค้น มาทำการคำนวณเพื่อหาตัวเลขสัมประสิทธิ์ความคล้ายคลึงกัน (Similarity Coefficient) หากเอกสารดังกล่าวประกอบไปด้วยคำที่ใกล้เคียงกันหรือตรงกันกับคำที่ต้องการสืบค้น จึงจะถือได้ว่าเป็น เอกสารที่ตรงประเด็น (Relevant Document) ดังแสดงหลักการพื้นฐานของแบบจำลองเวกเตอร์ในภาพที่ 2.2

ภาพที่ 2.2
แสดงเวกเตอร์ตัวแทนของเอกสารและเวกเตอร์ตัวแทนของคำที่ต้องการสืบค้น บนหลักการพื้นฐานของแบบจำลองเวกเตอร์



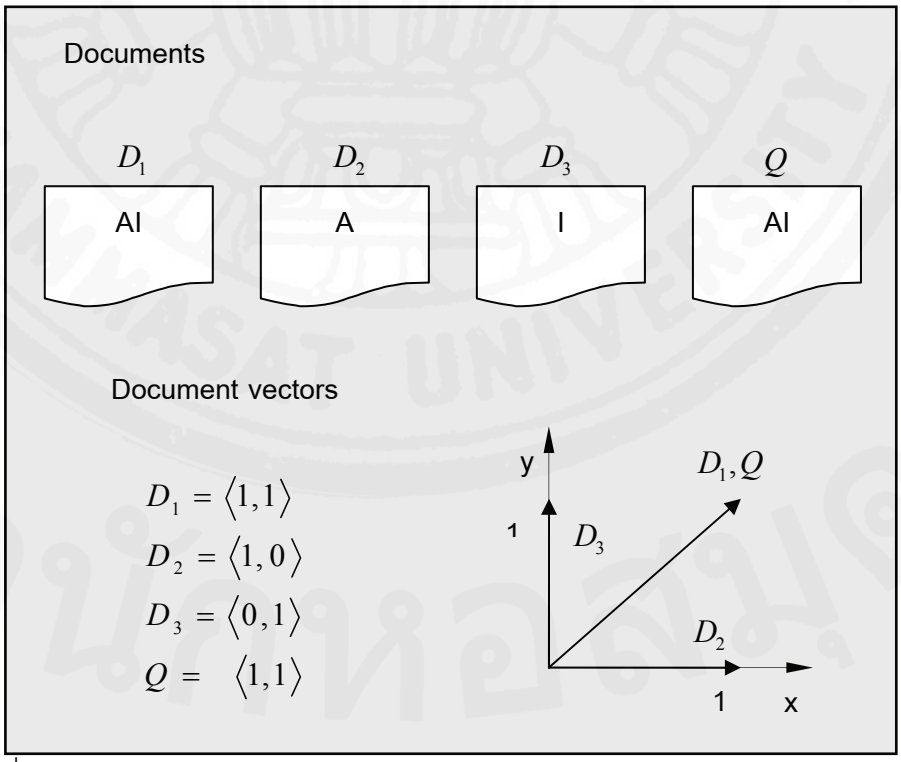
ที่มา: Grossman and Frieder (2004)

จากภาพที่ 2.2 แสดงเวกเตอร์ที่เป็นตัวแทนคำในแต่ละเอกสาร และเวกเตอร์ที่เป็นตัวแทนของคำที่ต้องการสืบค้น ซึ่งกระบวนการต่อไปจะนำตัวแทนเวกเตอร์ดังกล่าวไปทำการวัด

มุมระหว่างเวกเตอร์แต่ละเวกเตอร์ โดยใช้วิธีการคำนวณแบบ Dot Product และการคำนวณโดยใช้สูตรโคไซน์วัดค่าความคล้ายคลึงกัน (Cosine Similarity) เพื่อหาสัมประสิทธิ์ความคล้ายคลึงกัน (Similarity Coefficient)

พิจารณายกตัวอย่างโดยให้เอกสารที่รวบรวมไว้ประกอบไปด้วยองค์ประกอบ 2 ส่วน องค์ประกอบแรกเป็นตัวแทนการเกิดขึ้นของคำ α และองค์ประกอบที่สองเป็นตัวแทนการเกิดขึ้นของคำ β ซึ่งจะให้ค่าเป็น 1 เมื่อมีคำนั้นปรากฏในเวกเตอร์เอกสาร และให้ค่าเป็น 0 เมื่อคำนั้นไม่ปรากฏในเวกเตอร์เอกสาร ยกตัวอย่างเช่น เอกสารที่ 1 ประกอบไปด้วยการเกิดขึ้น 2 ครั้งสำหรับคำ α และไม่มีการเกิดขึ้นของคำ β ดังนั้นเวกเตอร์ตัวแทนเอกสารในตัวอย่างนี้ก็คือ $\langle 2,0 \rangle$ เป็นต้น แต่สำหรับเวกเตอร์ที่เป็นตัวแทนเอกสารใน 2 มิติ ที่มีสมาชิกของเวกเตอร์ดังนี้ $\langle 1,0 \rangle$ นั้นจะสามารถทำการคำนวณหาค่าความคล้ายคลึงกันได้ แต่ไม่สามารถที่จะนับจำนวนความถี่ของคำที่เกิดขึ้นภายในเอกสารได้

ภาพที่ 2.3
แบบจำลองเวกเตอร์ที่ประกอบไปด้วยคำศัพท์สองคำ



ที่มา: Grossman and Frieder (2004)

จากภาพที่ 2.3 แสดงตัวอย่างการสืบค้นตัวอักษรภาษาอังกฤษ 2 ตัวนั่นก็คือ A และ I โดยที่ตัวอักษรทั้งสองตัวนั้นถือได้ว่าเป็นองค์ประกอบหลักของเวกเตอร์ที่ต้องการค้นหา การแสดงเวกเตอร์ในมุมมอง 2 มิติ ทั้งเวกเตอร์ตัวแทนคำที่ต้องการสืบค้นและเวกเตอร์ตัวแทนเอกสารทั้ง 2 เอกสาร ทำให้สามารถหาค่าสัมประสิทธิ์ความคล้ายคลึงระหว่างคำที่ต้องการสืบค้น กับเอกสารทั้งหมดได้ด้วยการคำนวณระยะทางจากเวกเตอร์ตัวแทนคำที่ต้องการสืบค้น ไปจนถึงเวกเตอร์ตัวแทนเอกสารอีก 2 เวกเตอร์ และจากตัวอย่างนี้เวกเตอร์ตัวแทนเอกสารที่ 1 มีความคล้ายคลึงกันกับ เวกเตอร์ตัวแทนคำที่ต้องการสืบค้นมากที่สุด ดังนั้น การค้นคืนคำที่ต้องการค้นหาจากเอกสารที่ 1 จะตรงประเด็นมากที่สุดนั่นเอง

ในวิทยานิพนธ์ฉบับนี้ ทำการยกตัวอย่างการนำแบบจำลองเวกเตอร์ไปประยุกต์ใช้อย่างละเอียดทุกขั้นตอน เพื่อแสดงระบบการทำงานของเวกเตอร์คำ (Term Vector) ภายในแบบจำลองเวกเตอร์ ทั้งนี้โดยอาศัยเทคนิคและขั้นตอนจากเอกสารแนะนำการทำงานของเวกเตอร์คำแบบกระชับ (Term Vector Fast Track Tutorial) ของ Garcia (2006) มาเป็นแนวทางในการยกตัวอย่างแสดงการประยุกต์ใช้แบบจำลองเวกเตอร์ ดังต่อไปนี้

คำที่ต้องการสืบค้น ประกอบไปด้วยคำจำนวน 2 คำ นั่นคือคำว่า film และคำว่า movie จากเอกสารที่รวบรวมไว้มีสองคำนี้ปรากฏอยู่ประมาณ 200,000,000 เอกสาร จากนั้นทำการเลือกเอกสารที่ถูกเก็บรวบรวมไว้ ออกมา 2 เอกสารจาก 200,000,000 เอกสาร ดังนี้

คำว่า film ปรากฏในเอกสารที่ 1 (DOC1) 2 ครั้ง และปรากฏในเอกสารที่ 2 (DOC2) 1 ครั้ง ซึ่งเอกสารที่มีลักษณะดังกล่าวนี้ได้มีการรวบรวมไว้เป็นจำนวน 500,000 เอกสาร

คำว่า movie ปรากฏในเอกสารที่ 1 (DOC1) 3 ครั้ง และปรากฏในเอกสารที่ 2 (DOC2) 6 ครั้ง ซึ่งเอกสารที่มีลักษณะดังกล่าวนี้ได้มีการรวบรวมไว้เป็นจำนวน 600,000 เอกสาร

ขั้นตอนที่ 1 ทำการคำนวณเพื่อหาน้ำหนักของคำ ซึ่งสามารถหาได้จาก

$$w = tf \bullet \log \frac{N}{df_i} = tf \bullet \log(\frac{N-n}{n}) \qquad (2.1)$$

โดยที่ w คือ น้ำหนักของคำ (Term Weights)

tf คือ ความถี่ของคำ (Term Frequency) หรือ จำนวนครั้งที่คำปรากฏในเอกสาร

df_i คือ จำนวนเอกสารที่มีคำ Ti ปรากฏ โดยที่ i คือลำดับของคำที่ปรากฏในเอกสาร

IDF คือ ส่วนกลับของความถี่ของคำ (Inverse Document Frequency) โดยคำนวณได้จาก $\log(\frac{N-n}{n})$

โดยที่ N คือ จำนวนเอกสารทั้งหมด

n คือ จำนวนของเอกสารที่ปรากฏคำที่ต้องการค้นหา

คำนวณน้ำหนักของคำโดยสมการที่ (2.1) เพื่อทำการค้นหาว่า เอกสารที่ 1 (DOC1) หรือเอกสารที่ 2 (DOC2) ที่มีความใกล้เคียงกับคำสืบค้น (Query) มากที่สุด ซึ่งจะได้ค่าน้ำหนักของคำ จากการคำนวณ ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1

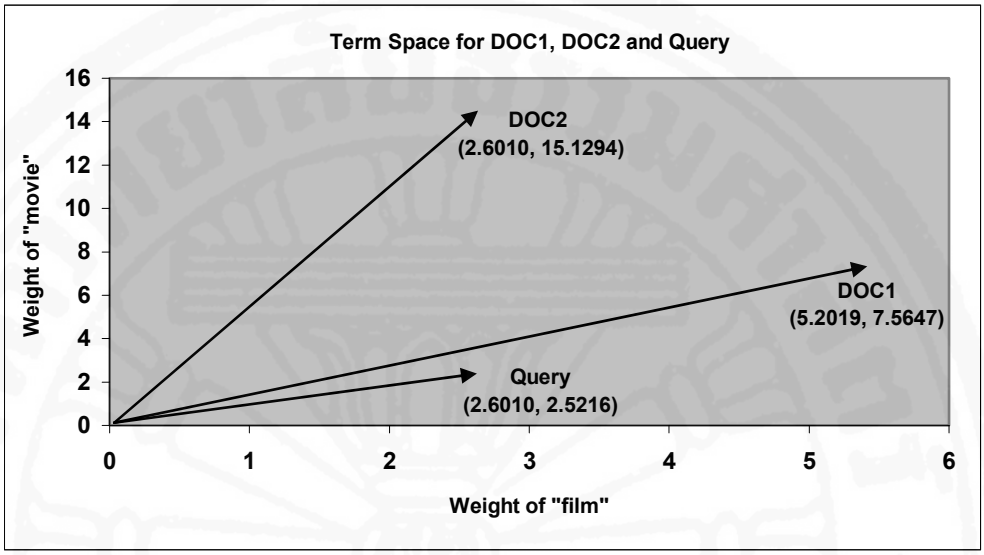
ตารางแสดงค่าน้ำหนักของคำที่ได้จากการคำนวณ

Item	Term	tf	N	n	IDF	w
DOC1	film	2	200,000,000	500,000	2.6010	5.2019
DOC1	movie	3	200,000,000	600,000	2.5216	7.5647
DOC2	film	1	200,000,000	500,000	2.6010	2.6010
DOC2	movie	6	200,000,000	600,000	2.5216	15.1294
Query	film	1	200,000,000	500,000	2.6010	2.6010
Query	movie	1	200,000,000	600,000	2.5216	2.5216

นำข้อมูลตัวเลขที่ได้ มาสร้างกราฟของ The film – movie Term Space โดยกำหนดให้น้ำหนักของคำว่า film และคำว่า movie เป็นแกน X และแกน Y ตามลำดับ และกำหนดให้จุดภายในกราฟเป็นคู่ลำดับของค่าน้ำหนักของคำในแต่ละเอกสารและในแต่ละ Query ดังแสดง ในภาพที่ 2.4

ภาพที่ 2.4

แสดงพื้นที่ของคำสำหรับ เอกสารที่ 1 (DOC1) เอกสารที่ 2 (DOC2) และ คำสืบค้น (Query)



ขั้นตอนที่ 2 ทำการคำนวณ Dot Products ระหว่างจุด

จากเวกเตอร์สองมิติ กำหนดเป็นตัวแปร X และ Y โดย Dot Products สามารถคำนวณได้โดย $(X_1 \times X_2) + (Y_1 \times Y_2)$ ได้ผลลัพธ์ดังแสดงในตารางที่ 2.2

ตารางที่ 2.2

แสดงการเปรียบเทียบ Dot Products ระหว่างเอกสารที่ 1 (DOC1) กับเอกสารที่ 2 (DOC2)

	Weight of gift term $X_1 \times X_2$	Weight of card term $Y_1 \times Y_2$	Result $(X_1 \times X_2) + (Y_1 \times Y_2)$
DOC1 ● Query	5.2019 × 2.6010	7.5647 × 2.5216	= 32.6053
DOC2 ● Query	2.6010 × 2.6010	15.1294 × 2.5216	= 44.9155

จากค่าที่แสดงในตารางที่ 2.2 แสดงให้เห็นว่า การพิจารณาเฉพาะผลที่ได้จากการคำนวณโดย Dot Products ระหว่างจุด พบว่าเอกสารที่ 2 (DOC2) มีค่าผลลัพธ์สูงกว่าเอกสารที่ 1

(DOC1) ซึ่งอาจทำให้ตั้งสมมติฐานไปได้ว่าเอกสารที่ 2 (DOC2) มีความใกล้เคียงกันกับคำสืบค้น (Query) มากกว่าเอกสารที่ 1 (DOC1)

ขั้นตอนที่ 3 คำนวณปริมาณทางเวกเตอร์ โดยใช้ Euclidean Distance

สำหรับเวกเตอร์สองมิติ ที่ต้องการพิจารณาระยะทางระหว่างจุด $P=(p_x,p_y)$ และ $Q=(q_x,q_y)$ โดยหลักการของ Euclidean Distance จะสามารถคำนวณระยะทางระหว่างจุดสองจุดที่มีพิกัดสองมิติดังกล่าว ได้จากสูตร

$$D=\sqrt{(p_x-q_x)^2+(p_y-q_y)^2} \qquad (2.2)$$

โดยที่ D คือ ระยะทางระหว่างจุด P และจุด Q

P คือ จุดที่พิจารณาใน The Term Space มีพิกัดสองมิติ

Q คือ จุดกำเนิด (Origin Coordinate)

ในกรณีนี้ การคำนวณปริมาณเวกเตอร์ ก็คือ การคำนวณระยะทางระหว่างจุดแต่ละจุดใน The Term Space ซึ่งมีพิกัดสองมิติ ดังแสดงในภาพที่ 2.5 กับ จุดเริ่มต้น (Origin Coordinate) การคำนวณทำได้โดยการนำค่าพิกัดแต่ละคู่อันดับ มาแทนลงในสมการของ Euclidean Distance สองมิติ ก็จะได้ผลการคำนวณปริมาณทางเวกเตอร์ดังแสดงในตารางที่ 2.3

ตารางที่ 2.3

แสดงผลการคำนวณปริมาณทางเวกเตอร์

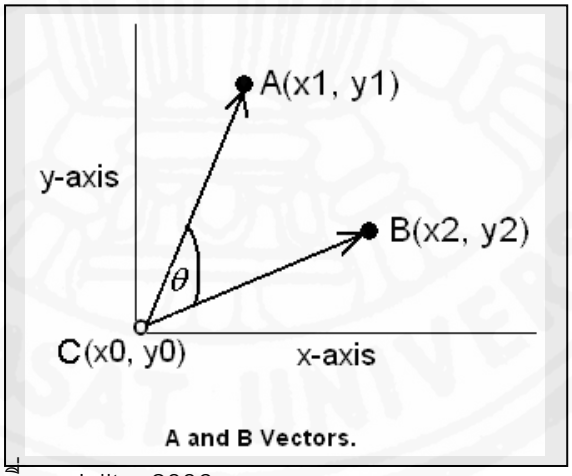
	p_x	p_y	q_x	q_y	$D=\sqrt{(p_x-q_x)^2+(p_y-q_y)^2}$
|DOC1|	5.2019	7.5647	0.0000	0.0000	9.1807
|DOC2|	2.6010	15.1294	0.0000	0.0000	15.3513
|Query|	2.6010	2.5216	0.0000	0.0000	3.6227

ขั้นตอนที่ 4 คำนวณค่าความคล้ายคลึงกัน (Similarity Measure) ระหว่างเอกสาร กับคำที่ต้องการสืบค้น โดยใช้สูตรค่าความคล้ายคลึงกันแบบโคไซน์ (Cosine Similarity) โดยการนำ Dot Products ที่ได้จากขั้นตอนที่ 2 มาทำการหารด้วย ผลลัพธ์ที่ได้จากการคูณปริมาณเวกเตอร์ในแต่ละเอกสาร กับปริมาณเวกเตอร์ของคำที่ต้องการสืบค้น ดังแสดงค่าแต่ละปริมาณเวกเตอร์ในขั้นตอนที่ 3 ซึ่งสรุปสุดท้ายแล้วจะได้ผลลัพธ์เป็นค่าความคล้ายคลึงกันของแต่ละเอกสารที่ถูกต้อง

โดยหลักการของ การคำนวณค่าความคล้ายคลึงกันแบบโคไซน์ (Cosine Similarity) ของเวกเตอร์สองเวกเตอร์ คือ เวกเตอร์ A และ เวกเตอร์ B นั้นกำหนดให้แต่ละเวกเตอร์มีพิกัดจุดปลายใน The Term Space ดังต่อไปนี้ คือ (X_1, Y_1) และ (X_2, Y_2) ตามลำดับ ดังแสดงในภาพที่ 2.5 และสูตรการหาค่าความคล้ายคลึงกันแบบโคไซน์ของเวกเตอร์สองเวกเตอร์ดังกล่าวแสดงในสมการที่ 2.3

ภาพที่ 2.5

แสดงเวกเตอร์ A และเวกเตอร์ B



ที่มา : Islita, 2006

สูตรการหาค่า Cosine ของเวกเตอร์ A และเวกเตอร์ B

$$Sim(A,B) = \cos ine\theta = \frac{A \bullet B}{|A||B|} = \frac{(X_1X_2)+(Y_1Y_2)}{(X_1^2+Y_1^2)^{\frac{1}{2}}(X_2^2+Y_2^2)^{\frac{1}{2}}} \quad (2.3)$$

โดยที่ $A \bullet B$ คือ Dot Products ระหว่างเวกเตอร์ A และเวกเตอร์ B

ในที่นี้คือ ผลลัพธ์ของ Dot Products ที่ได้จาก DOC1●Query และ DOC2●Query ตามลำดับ

|A||B| คือ Euclidean Distance ระหว่างเวกเตอร์ A และเวกเตอร์ B

ในที่นี้คือ ผลลัพธ์ของ Euclidean Distance ที่ได้จาก |DOC1|×|Query| และ |DOC2|×|Query| ตามลำดับ

โดยที่ ค่า Cosine ระหว่างมุมของเวกเตอร์จะมีค่าอยู่ระหว่าง 0 ถึง 1

ถ้าค่า Cosine มีค่าใกล้เคียงกับ 1 แสดงว่าเอกสารและคำสืบค้นมีความคล้ายคลึงกันมาก

ถ้าค่า Cosine มีค่าใกล้เคียงกับ 0 แสดงว่าเอกสารและคำสืบค้นมีความคล้ายคลึงกันน้อย

แสดงค่าโคไซน์แสดงความคล้ายคลึงกันระหว่างเอกสารแต่ละเอกสาร กับคำที่ต้องการสืบค้น ดังนี้

Cosine angle = DOC1 ● Query / (|DOC1|×|Query|)

= 32.6053 / (9.1807 × 3.6227)

= 0.9803

Cosine angle = DOC2 ● Query / (|DOC2|×|Query|)

= 44.9155 / (15.3513 × 3.6227)

= 0.8076

ขั้นตอนที่ 5 จากการคำนวณหาค่าความคล้ายคลึงกัน สามารถสรุปได้ว่าเอกสารที่ 1 (DOC1) มีความคล้ายคลึงกันกับคำสืบค้น (Query) มากกว่าเอกสารที่ 2 (DOC2)

ซึ่งในกรณีนี้หากไม่ทำการคำนวณค่าความคล้ายคลึงกัน การที่จะสังเกตข้อสรุปจากค่าของตัวเลขในขั้นตอนที่ 2 นั้น พบว่าค่าที่ได้จากการทำ Dot Product ของ DOC2 มีค่าสูงกว่า DOC1 ดังนั้นอาจสรุปไปได้ว่า DOC2 มีความคล้ายคลึงกันกับ Query มากกว่า DOC1 ซึ่งจากข้อสังเกตที่ได้นั้นไม่สอดคล้องกับข้อสรุปที่ได้จากการคำนวณค่าความความคล้ายคลึงกันที่ได้จากขั้นตอนที่ 5

แต่ในกรณีเดียวกันนี้ หากสังเกตภาพที่ได้จากขั้นตอนที่ 1 จะเห็นได้ว่า DOC2 มีความถี่ในการปรากฏคำว่า film ในเอกสารซ้ำกันถึง 6 ครั้ง ซึ่งนั่นเป็นสาเหตุทำให้ เวกเตอร์ของ DOC2 ห่างออกจาก เวกเตอร์ของ Query ซึ่งแสดงให้เห็นว่า ความถี่ของคำที่ปรากฏในเอกสารไม่มีส่วนในการค้นหาคำหรือเอกสารที่มีความคล้ายคลึงกัน ได้อย่างมีประสิทธิภาพ

กล่าวโดยสรุป จากตัวอย่างการนำแบบจำลองเวกเตอร์ ไปประยุกต์ใช้ดังกล่าว แสดงให้เห็นถึง ขั้นตอนการสร้างเวกเตอร์คำ (Term Vector) และเวกเตอร์เอกสาร (Document Vector) เพื่อเป็นตัวแทนของคำและเอกสารภายในแบบจำลองเวกเตอร์ ตามลำดับ ดังเช่น

ภาพที่ 2.6
เมตริกซ์ของคำและเอกสาร

$$\begin{pmatrix} & D_1 & D_2 & \dots & D_n \\ T_1 & W_{1,1} & W_{1,2} & \dots & W_{1,n} \\ T_2 & W_{2,1} & W_{2,2} & \dots & W_{2,n} \\ . & . & . & \dots & . \\ . & . & . & \dots & . \\ T_t & W_{t,1} & W_{t,2} & \dots & W_{t,n} \end{pmatrix}$$

ซึ่งแสดงให้เห็นว่า เมตริกซ์ของคำและเอกสาร (Term – Document Matrix) ดังแสดงในภาพที่ 2.6 สามารถเป็นตัวแทนของเอกสารภายในแบบจำลองเวกเตอร์ได้ โดยมีจำนวนเอกสารที่ทำการเก็บรวบรวมไว้ทั้งหมด n เอกสาร และสมาชิกภายในเมตริกซ์คือ น้ำหนักของคำในเอกสาร หากค่าของน้ำหนักของคำที่ได้มีค่าเป็นศูนย์ แสดงว่าคำๆ นั้นไม่มีนัยสำคัญกับเอกสาร หรือคำๆ นั้นไม่ปรากฏอยู่ในเอกสาร เป็นต้น

จากตัวอย่างการประยุกต์ใช้แบบจำลองเวกเตอร์ดังกล่าว แสดงให้เห็นว่า การนำแบบจำลองเวกเตอร์มาใช้ในศาสตร์การค้นคืนข้อมูล นั้นจะต้องอาศัยทฤษฎีและเทคนิคหลากหลาย ดังจะกล่าวในหัวข้ออื่นๆ ต่อไป

2.1.1.2 การตัดคำ

การตัดคำ (Word Segmentation) คือ การแบ่งข้อความที่ต่อเนื่องกันออกเป็นหน่วยคำ (Morpheme) หรือลักษณะของการรู้จำ (Recognition) คำๆ หนึ่งในข้อความที่ต่อเนื่องกัน จุดประสงค์ของการตัดคำ นั้นก็คือการแบ่งสายอักขระของตัวอักษร (String) เพื่อหาขอบเขตของแต่ละหน่วยคำ ซึ่งวิธีการที่จะนำมาใช้ในการตัดคำนั้นขึ้นอยู่กับลักษณะเฉพาะของภาษานั้นๆ (วิรัช ศรเลิศล้ำวาณิช, 2536)

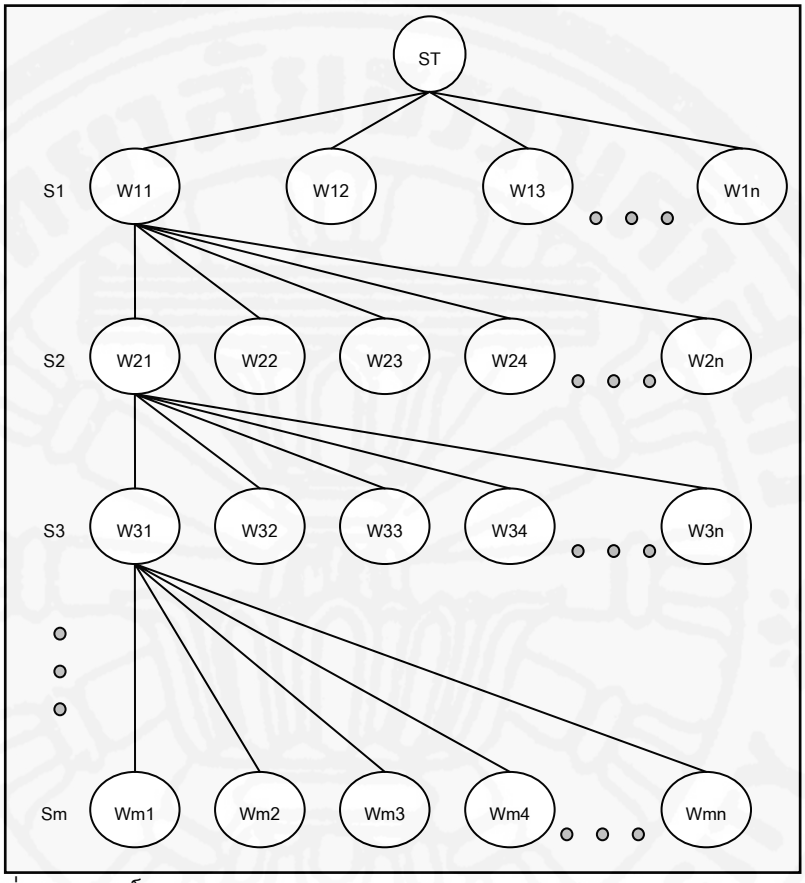
เนื่องจากวิทยานิพนธ์ฉบับนี้มีจุดประสงค์ในการทำวิจัยเกี่ยวข้องกับภาษาไทย ดังนั้นจึงมีแนวทางการศึกษาด้าน การตัดคำภาษาไทย (Thai Word Segmentation) ซึ่งเมื่อพิจารณาลักษณะเฉพาะของภาษาไทยแล้ว พบว่า เป็นภาษาที่มีการเขียนคำติดต่อกัน ไม่มีการเว้นวรรคระหว่างคำ ไม่มีการใช้เครื่องหมายวรรคตอนที่ชัดเจน ไม่มีตัวบ่งชี้หน้าที่ทางไวยากรณ์ (Syntactic Marker Postposition) ไม่มีการแปรรูป (Inflection) และไม่มีการใช้ตัวอักษรใหญ่ตัวอักษรเล็ก ลักษณะพื้นฐานของประโยคภาษาไทย จะประกอบไปด้วยคำ หรือกลุ่มของคำหลายๆ คำ (พิสิทธิ์ พรมจันทร์, 2540) ได้อธิบายรายละเอียดและมีการกำหนดค่าดังภาพที่ 2.7

หากทำการตัดคำประโยคภาษาไทย ยกตัวอย่างประโยค “ฉันมารอกราบ” เมื่อผ่านการตัดคำแล้วจะได้ประโยคดังนี้ “ฉัน | มา | รอ | กราบ” “ฉัน | มา | รอก | ราบ” และ “ฉัน | มาร | อก | ราบ” ซึ่งถือได้ว่าเป็นปัญหาที่สำคัญสำหรับขั้นตอนวิธีการตัดคำ ที่ทำให้ระดับความถูกต้องของประโยค (Syntax Validity) ที่ได้หลังจากการตัดคำแตกต่างกันไป

จากลักษณะเฉพาะของภาษาไทยดังที่กล่าวมาแล้ว วิทยานิพนธ์ฉบับนี้จึงได้นำหลักการการตัดคำไทยในระบบแปลภาษา (Word Segmentation for Thai in Machine Translation System) (วิรัช ศรเลิศล้ำวาณิช, 2536) มานำเสนอหลักการการตัดคำที่เหมาะสมกับลักษณะเฉพาะของภาษาไทย ซึ่งวิธีการดังกล่าวได้นำอัลกอริทึมสำหรับการตัดคำโดยใช้พจนานุกรมมาประยุกต์ใช้กับภาษาไทย จำนวน 3 อัลกอริทึม ดังจะกล่าวต่อไป

ภาพที่ 2.7

แสดงลักษณะพื้นฐานของประโยคภาษาไทย



ที่มา: พิสิทธิ์ พรมจันทร์ (2540)

ST แทนประโยค

Si แทนกลุ่มของคำ

i แทนลำดับที่ของกลุ่มคำ

S1, S2, S3, …, Sm คือกลุ่มของคำที่ปรากฏในประโยคลำดับที่ 1, 2, 3, …, m ตามลำดับ

Wij แทนคำศัพท์ที่เป็นสมาชิกในกลุ่มคำศัพท์ Si ลำดับที่ j

จะได้ว่า ST = [S1, S2, S3, …, Sm-1, Sm]

เมื่อ Si = [Wi1, Wi2, …, Win]

กฎทางอักขระวิธี

กฎทางอักขระวิธี คือ วิธีที่กำหนดลักษณะของการประสมอักษร การเว้นวรรคตามกฎเกณฑ์ที่ชัดเจน ก็จะสามารถช่วยลดความคลุมเครือของคำหรือข้อความได้ และการขึ้นย่อหน้า ซึ่งทั้ง 3 ลักษณะดังกล่าวมีส่วนช่วยในการบ่งชี้ขอบเขตของการพิจารณาเพื่อเปรียบเทียบกับคำในพจนานุกรม โดย

- การขึ้นย่อหน้า เป็นตัวบ่งชี้ได้ดีถึงการสิ้นสุดของข้อความ
- การเว้นวรรค เป็นตัวบ่งชี้ถึงความเป็นไปได้ของการสิ้นสุดของคำหรือประโยค
- กฎทางอักขระวิธี เป็นตัวบ่งชี้ถึงความเป็นไปได้ของการพิจารณาแยกสายอักขระออกเพื่อการพิจารณาคำ

ฉะนั้นจึงมีการแบ่งอักขระออกเป็น 5 กลุ่มดังต่อไปนี้

อักขระกลุ่มที่ 1 คือ กลุ่มอักขระที่เมื่อผสมเข้ากันกับพยัญชนะใดๆ แล้วไม่ทำให้มีการเคลื่อนขวาของตำแหน่งที่จะเขียนต่อไป และอักขระกลุ่มนี้จะไม่สามารถเขียนอยู่ในตำแหน่งเดี่ยวๆ ได้ หรือที่เรียกกันว่า Non-spacing Character กลุ่มอักขระกลุ่มนี้ประกอบไปด้วยอักขระที่อยู่ในรูปสระ วรรณยุกต์ และเครื่องหมายพิเศษ นั่นคือ ◌ั, ◌ิ, ◌ี, ◌ึ, ◌ื, ◌ุ, ◌ู, ◌็, ◌่, ◌้, ◌๊, ◌๋, ◌์, ◌ำ, ◌ฺ

อักขระกลุ่มที่ 2 คือ กลุ่มอักขระที่จำเป็นต้องมีรูปพยัญชนะตามเสมอ กลุ่มอักขระกลุ่มนี้ประกอบไปด้วย เ_, แ_, โ_, ใ_, ไ_

อักขระกลุ่มที่ 3 คือ กลุ่มอักขระที่จำเป็นต้องมีรูปพยัญชนะนำหน้าเสมอ กลุ่มอักขระกลุ่มนี้ประกอบไปด้วย _ะ, _า, _ำ, _ๅ, โดยสำหรับอักขระ _ๅ จะมีพยัญชนะนำหน้าได้เพียง 2 พยัญชนะ คือ ฤ และ ฦ

อักขระกลุ่มที่ 4 คือ กลุ่มของอักขระที่มีไม้ทัณฑฆาตบังคับข้างบนหรือมีตัวการันต์ ◌์ โดยตัวการันต์เป็นพยัญชนะตัวสุดท้ายที่ไม่มีการอ่านออกเสียง

อักขระกลุ่มที่ 5 คือ กลุ่มของอักขระที่เหลือทั้งหมด

เนื่องจากข้อมูลสายอักขระจัดเก็บไว้ในชุดรหัส X-TIS (eXtended Thai Industrial Standard code set, X-TIS) ซึ่งเป็นการกำหนดรหัสในไบท์ที่ 2 เนื่องจากว่ารหัสของ Non-Spacing Character หรืออักขระกลุ่มที่ 1 ถูกเก็บรวมเข้ากับพยัญชนะที่ถูกประสม และมีตำแหน่งในระบบรหัสที่สัมพันธ์กับการแสดงผลบนจอไปแล้ว จึงไม่ต้องคำนึงถึง ดังนั้นจะมีเพียง อักขระกลุ่มที่ 2, 3 และ 4 เป็นตัวจำกัดที่ดีที่จะไม่ให้มีการพิจารณาการตัดคำในระหว่างตัวอักขระนั้นๆ (วิรัช ศรเลิศล้ำวาณิช, 2536)

วิธีการเทียบคำยาวที่สุด

วิธีการเทียบคำยาวที่สุด (Longest Matching) นั้นถือได้ว่าเป็น วิธีการทางฮิวริสติก (Heuristic) วิธีการหนึ่ง เนื่องจากว่าภาษาไทยไม่มีเครื่องหมายบอกจุดสิ้นสุดของข้อความ ดังนั้นจากกฎอักขระวิธีที่กล่าวมาแล้วนั้น จะเป็นเกณฑ์ที่ช่วยบ่งบอกถึงขอบเขตของหน่วยของคำ หรือหน่วยของข้อความได้ โดยทำการตรวจสอบคำเริ่มตั้งแต่หน่วยของคำที่สั้นที่สุดที่สามารถคัดมาได้ จากนั้นนำหน่วยของคำที่คัดมาได้มาทำการตรวจสอบกับคำในพจนานุกรมแล้วแยกให้เป็นคำที่ตรงกับพจนานุกรมเรียงจากซ้ายไปขวา เมื่อตรวจสอบกับคำในพจนานุกรมแล้วหากไม่พบก็ทำการลดความยาวของข้อความลงทีละตัว ตามกฎอักขระวิธี วิธีการนี้ให้ความถูกต้องในการตัดคำประมาณร้อยละ 80

งานวิจัยเรื่อง การตัดคำไทยในระบบแปลภาษา ของ วิรัช ศรเลิศล้ำวาณิช (2536) ได้ทำการยกตัวอย่างการตัดคำโดยวิธีการตัดคำแบบเลือกคำที่ยาวที่สุด ภายใต้เงื่อนไขที่ว่า เมื่อตรวจสอบกับคำในพจนานุกรมแล้วไม่พบ ก็ให้ทำการลดความยาวของข้อความนั้นๆ ลงทีละตัว โดยพิจารณาจากกฎเกณฑ์ทางอักขระวิธีดังที่อธิบายไว้แล้ว ยกตัวอย่างประโยคเช่น “ความก้าวหน้าทางวิทยาศาสตร์มีบทบาทสำคัญ” แสดงผลของการตัดคำโดยวิธีการตัดคำแบบเลือกคำที่ยาวที่สุดภายใต้เงื่อนไขกฎเกณฑ์ทางอักขระวิธี ดังต่อไปนี้

ตารางที่ 2.4
แสดงการตัดคำแบบเลือกคำที่ยาวที่สุด

ประโยค	ส่วนของคำที่ยาวที่สุด
ความก้าวหน้าทางด้านวิทยาศาสตร์มีบทบาทสำคัญ	-
ทางด้านวิทยาศาสตร์มีบทบาทสำคัญ	ความก้าวหน้า
ด้านวิทยาศาสตร์มีบทบาทสำคัญ	ทาง
วิทยาศาสตร์มีบทบาทสำคัญ	ด้าน
มีบทบาทสำคัญ	วิทยาศาสตร์
บทบาทสำคัญ	มี
สำคัญ	บทบาท
-	สำคัญ

ที่มา: วิรัช ศรเลิศล้ำวาณิช (2536)

โดยสรุปแล้วจากการตัดคำโดยวิธีการตัดคำแบบเลือกคำที่ยาวที่สุด สามารถตัดคำในประโยค “ความก้าวหน้าทางด้านวิทยาศาสตร์มีบทบาทสำคัญ” ได้เป็น “ความก้าวหน้า | ทาง | ด้าน | วิทยาศาสตร์ | มี | บทบาท | สำคัญ” นั่นเอง

งานวิจัยเรื่อง การตัดคำภาษาไทยโดยใช้คุณลักษณะ ของ ไพศาล เจริญพรสวัสดิ์ (2541) ได้กล่าวถึงการยกตัวอย่างการตัดคำโดยวิธีการตัดคำแบบเลือกคำที่ยาวที่สุด ขึ้นมาอีกตัวอย่างหนึ่ง ดังแสดงในตารางที่ 2.4 ทั้งนี้เนื่องจากเมื่อนำพจนานุกรมเข้ามาพิจารณาในการตัดคำแล้ว พบว่าความถูกต้องในการตัดคำเพิ่มขึ้น และการตัดคำโดยพจนานุกรมมีความถูกต้องในการตัดคำมากกว่าการตัดคำโดยใช้กฎเพียงอย่างเดียว โดยรายละเอียดขั้นตอนวิธีการตัดคำนั้นจะคล้ายคลึงกันกับวิธีการแบ่งพยางค์ด้วยพจนานุกรม แตกต่างกันตรงที่การแบ่งคำด้วยพจนานุกรมนั้นจัดเก็บคำลงในพจนานุกรมแทนพยางค์ แต่ส่วนขั้นตอนวิธีการทำงานเหมือนเดิมคือใช้ขั้นตอนวิธีการตัดคำแบบเลือกคำที่ยาวที่สุด ดังที่ได้กล่าวไปแล้วนั่นเอง

ตารางที่ 2.5
แสดงการตัดคำแบบเลือกคำที่ยาวที่สุด และแสดงขั้นตอนการย้อนรอย (Backtracking)

ประโยค	คำที่ได้	คำที่ถูกเลือก
โคนมนอนบนกองหญ้า	โค,โคน	โคน
มนอนบนกองหญ้า	-	(ย้อนรอย)
โคนมนอนบนกองหญ้า	โค,โคน	โค (เลือกคำรองลงมา)
นมนอนบนกองหญ้า	นม	นม
นอนบนกองหญ้า	นอ, นอน	นอน
บนกองหญ้า	บน	บน
กองหญ้า	กอ, กอง	กอง
หญ้า	หญ้า	หญ้า

ที่มา: ไพศาล เจริญพรสวัสดิ์ (2541)

โดยสรุปแล้วจากการตัดคำโดยวิธีการตัดคำแบบเลือกคำที่ยาวที่สุด สามารถตัดคำในประโยค “โคนมนอนบนกองหญ้า” ได้เป็น “โค | นม | นอน | บน | กอง | หญ้า” นั่นเอง

วิธีการตัดคำให้ได้จำนวนคำและคำที่ไม่มีในพจนานุกรมน้อยที่สุด

วิธีการตัดคำให้ได้จำนวนคำและคำที่ไม่มีในพจนานุกรมน้อยที่สุด (Unknown Word)นี้ถือได้ว่าเป็นวิธีการทางฮิวริสติกอีกวิธีการหนึ่ง โดยจะเป็นวิธีการที่สามารถช่วยได้มากในกรณีที่ข้อความประกอบไปด้วยคำจำนวนมาก หรือมีความยาวของข้อความมาก เนื่องจากวิธีการนี้จะทำบนอัลกอริทึมการเทียบคำยาวที่สุด ดังที่กล่าวมาแล้ว ซึ่งอัลกอริทึมนี้จะทำการหาความเป็นไปได้ทั้งหมดในการตัดคำสำหรับข้อความนั้นๆ อีกทั้งยังมี วิธีการถอยกลับ (Backtracking) ซึ่งจะเริ่มกระทำหลังจากที่ได้คำตอบจากอัลกอริทึมการเทียบคำยาวที่สุด แล้วโดยจะกระทำทีละคำจากซ้ายไปขวา เมื่อทำการถอยกลับ ครบทุกคำแล้ว ก็จะทำการคำนวณหาค่า Cost ของแต่ละส่วนของคำที่เป็นไปได้ ยกตัวอย่างเช่น “กีฬา | เป็นการ | ออกกำลัง | กาย | อย่างหนึ่ง” มีค่า Cost เท่ากับ 5 เป็นต้น

งานวิจัย การตัดคำภาษาไทยโดยใช้คุณลักษณะ ของ ไพศาล เจริญพรสวัสดิ์ นำเสนอ งานวิจัยที่เกี่ยวข้องเรื่อง การตัดคำภาษาไทย ของ วิรัช ศรเลิศล้ำวาณิช ซึ่งได้กล่าวถึง การตัดคำโดยเลือกแบบเหมือนมากที่สุด (Maximal Matching) จากการอ้างอิงและจากการศึกษาหลักการแล้วพบว่า วิธีการตัดคำให้ได้จำนวนคำและคำที่ไม่มีในพจนานุกรมน้อยที่สุด นี้มีหลักการเดียวกันกับ วิธีการตัดคำโดยเลือกแบบเหมือนมากที่สุด และงานวิจัยการตัดคำภาษาไทยโดยใช้คุณลักษณะ ได้กล่าวถึงการตัดคำโดยเลือกแบบเหมือนมากที่สุด ไว้ว่าเป็นขั้นตอนวิธีที่สามารถแก้ไขการตัดคำแบบเลือกคำยาวที่สุด ได้ โดยแก้ไขในส่วนที่การตัดคำแบบเลือกคำยาวที่สุดนั้นเลือกคำที่มีความยาวมากเกินไปตั้งแต่แรก ทำให้ข้อความที่ตามมามีความผิดพลาดไปโดยได้ง่าย

ยกตัวอย่างประโยค “ไปหามเหสี”
สามารถตัดคำได้เป็น “ไป | หาม | เห | สี” กับ “ไป | หา | มเหสี”
เลือกประโยคที่มีการตัดคำแล้วได้ความหมายดังนี้ “ไป | หา | มเหสี”

โดยวิธีการตัดคำโดยเลือกแบบเหมือนมากที่สุด นี้จะเลือกประโยค “ไป หา มเหสี” ซึ่งเป็นประโยคที่ให้ความหมายถูกต้องและเหมาะสม

ในกรณีที่ทำการตัดคำแล้วได้จำนวนคำที่เท่ากัน ให้นำการตัดแบบเลือกคำยาวที่สุดเข้ามาช่วยพิจารณา

ยกตัวอย่างประโยค “ฉันนั่งตากลม”
สามารถตัดคำได้เป็น “ฉัน | นั่ง | ตาก | ลม” และ “ฉัน | นั่ง | ตา | กลม”
หมายเหตุ: มีจำนวนคำที่ตัดมาแล้วเท่ากันทั้ง 2 ประโยค
เลือกประโยคที่มีการตัดคำแล้วได้ความหมายดังนี้ “ฉัน | นั่ง | ตาก | ลม”

ดังนั้นเมื่อใช้การตัดคำแบบเลือกคำยาวที่สุด เข้ามาช่วยพิจารณา จะเลือกประโยค “ฉัน นั่ง ตาก ลม” ซึ่งเป็นประโยคที่ได้ทำการเลือกมาโดยการประยุกต์วิธีการตัดคำโดยเลือกแบบเหมือนมากที่สุด และ การตัดคำแบบเลือกคำยาวที่สุด มาพิจารณาร่วมกันจึงได้ผลลัพธ์เป็นประโยคที่ให้ความหมายถูกต้องและเหมาะสม

โดยหลักการการตัดคำไทยในระบบแปลภาษา ได้นำอัลกอริทึมทั้ง 3 อัลกอริทึมมาใช้ ถึงแม้ว่าหลักการ และกระบวนการทำงานของอัลกอริทึม ดังกล่าว ไม่ได้ทำการวิเคราะห์ในเชิงลึก

ถึงโครงสร้างทางไวยากรณ์ หรือความสัมพันธ์ทางความหมายของคำหรือข้อความ แต่ในเชิงปฏิบัติ เมื่อนำหลักการการทำงานของอัลกอริทึมทั้ง 3 อัลกอริทึมประกอบเข้าด้วยกันแล้ว พบว่าผลของการตัดคำที่ได้จะให้ความถูกต้องสูงถึงร้อยละ 90 และให้ผลของการตัดคำเร็วพอที่จะใช้งานในขั้นตอนต่อไปได้ อีกทั้งเป็นวิธีที่ใช้ได้ผลดี และในการพัฒนาระบบเครื่องแปลภาษาผลของการใช้งานจริงนั้น เป็นตัวกระตุ้นให้มีการวิจัยและพัฒนาโปรแกรมการตัดคำและกำกับหน้าที่คำภาษาไทยอัตโนมัติ มาจนถึงรูปแบบ (Version) ที่ใช้กันอยู่ในปัจจุบันนี้ (วิรัช ศรเลิศล้ำวาณิช, 2536)

เนื่องด้วยเหตุผลดังกล่าว งานวิจัยนี้จึงเลือกใช้ โปรแกรมตัดคำและกำกับหน้าที่คำภาษาไทยอัตโนมัติ (Smart Word Analysis for Thai, SWATH) หรือที่เรียกกันโดยทั่วไปว่า โปรแกรมสว็อท ที่ถูกพัฒนาขึ้นบนพื้นฐานของอัลกอริทึมสำหรับการตัดคำโดยใช้พจนานุกรมแบบวิธีการตัดคำแบบเลือกคำยาวที่สุด และวิธีการตัดคำโดยเลือกแบบเหมือนมากที่สุด โดยที่โปรแกรมดังกล่าว นำมาใช้ในงานวิจัยนี้เพื่อทำการแบ่งขอบเขตของแต่ละคำ เพื่อที่จะให้สามารถดำเนินการจัดการกับคำหรือข้อความนั้นๆ ได้อย่างสะดวกและถูกต้อง เพื่อที่จะให้กระบวนการทำงานต่อเนื่องในระดับที่ลึกกว่าสามารถทำงานต่อไปได้ และเพื่อกำหนดขอบเขตของคำที่ใช้สำหรับทำการวิเคราะห์ในระบบต่อไปได้ ซึ่งระบบการประเมินการเขียนเรียงความภาษาไทยที่นำเสนอในวิทยานิพนธ์ฉบับนี้ มีความจำเป็นอย่างยิ่งที่จะต้องหาขอบเขตของคำ เพื่อที่จะวิเคราะห์หาความหมายแอบแฝง ด้วยกระบวนการแยกค่าแบบเดี่ยว ซึ่งเป็นกระบวนการในระดับลึกของระบบ ให้สามารถทำงานต่อไปได้

2.1.1.3 **น้ำหนักของคำ**

เมื่อกล่าวถึง แบบจำลองเวกเตอร์ กลุ่มของคำภายในเอกสารที่มีความคล้ายคลึงกัน (Intra Clustering Similarity) นั้นถูกกำหนดค่าความคล้ายคลึงของคำภายในเอกสารเหล่านั้นได้โดยการวัดค่าความถี่เบื้องต้นของคำที่ปรากฏอยู่ภายในเอกสาร ซึ่งความถี่ของคำที่ปรากฏอยู่ภายในเอกสารนี้จะถูกอ้างอิงโดยปัจจัยที่เรียกว่า ค่าความถี่ของคำที่ปรากฏอยู่ในเอกสาร (Term Frequency, tf factor) นอกจากนี้ยังมี กลุ่มของคำภายในเอกสารที่ไม่มีความคล้ายคลึงกัน (Intra Clustering Dissimilarity) ถูกกำหนดความไม่คล้ายคลึงกันโดยการนำค่าความถี่ของคำที่ปรากฏอยู่ภายในเอกสารทั้งหมดมาทำการแปรผกผัน โดยทั่วไปจะเรียกปัจจัยนี้ว่า ส่วนกลับของค่าความถี่เอกสาร (Inverse Document Frequency, idf factor) มีการใช้ ส่วนกลับของค่าความถี่

เอกสารในการระบุคำซึ่งปรากฏอยู่ในหลากหลายเอกสารแต่เป็นคำที่ไม่เป็นประโยชน์ต่อการจำแนกเอกสารที่ตรงประเด็นได้นั้นเอง ทั้งนี้ได้มีการนำเสนออัลกอริทึมการจัดกลุ่มของคำภายในเอกสารที่มีประสิทธิภาพ เพื่อทำการวัดความเหมาะสมของคำที่จะเป็นตัวแทนของเอกสารหรือเนื้อหาภายในเอกสารนั้นๆ โดยมีนิยามของปัจจัยต่างๆ ดังนี้

คำนิยามน้ำหนักของคำ มีดังนี้คือ กำหนดให้ N คือ จำนวนเอกสารทั้งหมดที่อยู่ในระบบ n_i คือ จำนวนเอกสารที่ปรากฏคำที่พิจารณา $freq_{i,j}$ คือ ค่าความถี่ของคำที่ปรากฏในเอกสาร k_i คือ คำที่พิจารณาการปรากฏในเอกสาร d_i คือ เอกสารพิจารณาการปรากฏของคำ และ $f_{i,j}$ คือ ค่าความถี่โดยทั่วไป (Normalized Frequency) ของคำที่ปรากฏอยู่ในเอกสาร เพราะฉะนั้น จากคำนิยามตัวแปรต่างๆ ดังกล่าว ทำให้สามารถหา ค่าความถี่โดยทั่วไป ของคำที่ปรากฏอยู่ในเอกสาร ได้ดังสมการต่อไปนี้

$$f_{i,j} = \frac{freq_{i,j}}{\max_l freq_{l,j}} \qquad (2.4)$$

โดยที่ ค่าสูงสุด นั้นได้จากการเปรียบเทียบค่าความถี่ของคำทุกคำที่ปรากฏอยู่ในเอกสาร แล้วเลือกคำที่มีค่าความถี่สูงที่สุดมาแทนค่า $\max_l freq_{l,j}$ ดังแสดงในสมการที่ 2.4 จากนั้นจึงนำค่าดังกล่าวมาคำนวณต่อจนได้ค่าความถี่โดยทั่วไปของคำที่ปรากฏในเอกสารที่มีมาตรฐาน นั่นเอง แต่หากคำ k_i ไม่ปรากฏในเอกสาร d_i จะส่งผลให้ $f_{i,j} = 0$

นอกจากนี้จะนำเสนอ ส่วนกลับของค่าความถี่ของเอกสาร โดยสามารถคำนวณค่าดังกล่าวได้จากสมการดังต่อไปนี้

$$idf_i = \log \frac{N}{n_i} \qquad (2.5)$$

นิยามของ น้ำหนักของคำ นั้นมีอยู่หลากหลายรูปแบบ แต่อย่างไรก็ตามสูตรการคำนวณหาน้ำหนักของคำ ที่นิยมใช้กันโดยทั่วไป (Yates and Neto, 1999) นั่นก็คือ

$$w_{i,j} = f_{i,j} \times \log \frac{N}{n_i} \qquad (2.6)$$

โดยที่สูตรการคำนวณหาน้ำหนักของคำนี้อาจจะมีรูปแบบตัวแปรที่ใช้ในการคำนวณที่แตกต่างกันได้ ทั้งนี้เนื่องจากมีการนำหลักการที่เรียกกันโดยทั่วไปว่า tf – idf มาประยุกต์ใช้ในการหาน้ำหนักของคำ จากสูตรพื้นฐานดังกล่าวสามารถนำมาประยุกต์เพื่อใช้ในการคำนวณสำหรับหาน้ำหนักของคำจากข้อมูลที่หลากหลายได้ดังต่อไปนี้

$$w_{i,j} = (0.5 + \frac{0.5\, freq_{i,q}}{\max_l freq_{l,q}}) \times \log \frac{N}{n_i} \qquad (2.7)$$

โดยที่ $freq_{i,q}$ คือ ค่าความถี่ของคำ k_i ที่ปรากฏในเอกสารที่มีข้อมูลต้องการค้นหา q

$\max_l freq_{l,q}$ คือ ค่าความถี่สูงที่สุดของคำที่ปรากฏในเอกสารที่มีข้อมูลต้องการค้นหา q

2.1.1.4 **พีชคณิตเชิงเส้นสำหรับศาสตร์การค้นคืนข้อมูล**

พีชคณิตเชิงเส้นสำหรับศาสตร์การค้นคืนข้อมูล (Linear Algebra for Intelligent Information Retrieval) กล่าวถึง กระบวนการแยกค่าแบบเดี่ยว (Singular Value Decomposition) ว่าเป็นกระบวนการที่ใช้หลักการทางพีชคณิตเชิงเส้นในการแยกตัวประกอบของเมตริกซ์ที่มีความหมายสำคัญ ด้วยการประยุกต์ใช้ขบวนการที่หลากหลายร่วมกันกับหลักการทางสถิติศาสตร์ อีกทั้งกระบวนการแยกค่าแบบเดี่ยวนั้นยังสามารถนำมาประยุกต์ใช้กับเมตริกซ์ได้ทุกรูปแบบอีกด้วย

หลักการพื้นฐานโดยทั่วไปของ กระบวนการแยกค่าแบบเดี่ยว จะถูกนำมาใช้ในการแก้ปัญหาสมการเส้นตรงยกกำลังสองน้อยที่สุด (Linear Least Squares) การจัดอันดับเมตริกซ์ (Matrix Rank Estimation) และการวิเคราะห์ความสัมพันธ์ (Correlation Analysis) โดยกำหนดให้ เมตริกซ์ A มีขนาดเมตริกซ์ $m \times n$ และมีขนาดมิติของเมตริกซ์โดยทั่วไปเป็น $m \geq n$ อันดับของเมตริกซ์ A มีค่าเท่ากับค่า r ในที่นี้กระบวนการแยกค่าแบบเดี่ยวของเมตริกซ์ A เขียนแทนด้วย SVD (A) ดังจะกล่าวถึงองค์ประกอบของเมตริกซ์ A และทฤษฎีที่เกี่ยวข้องที่สามารถคำนวณหาเมตริกซ์ A_k ได้ดังต่อไปนี้

ทฤษฎีที่ 1 กำหนดให้ SVD ของเมตริกซ์ A มีการคำนวณดังสมการต่อไปนี้

$$A = U\Sigma V^T \tag{2.8}$$

โดย $U^T U = V^T V = I_n$

$$\Sigma = diag(\sigma_1, ..., \sigma_n)$$

$$\sigma_i > 0 \, for 1 \le i \le r$$

$$\sigma_j = 0 \, for \;\; j \ge r+1$$

และ $\sigma_1 \ge \sigma_2 \ldots \ge \sigma_r > \sigma_{r+1} = \ldots = \sigma_n = 0$ (Berry, Dumis and Brien, 1994)

การแยกค่าแบบเดี่ยวของเมตริกซ์ A ขนาดเมตริกซ์ $m \times n$ โดยที่ $m \ge n$ แยกค่าแบบเดี่ยวออกเป็น 3 เมตริกซ์นั่นก็คือ เมตริกซ์ U มีขนาด $m \times r$, เมตริกซ์ $\sum$ มีขนาด $r \times r$ และเมตริกซ์ V มีขนาด $r \times n$ โดยที่ค่า r คืออันดับของเมตริกซ์ A (Rank (A) = r)

เมตริกซ์ $\sum$ คือเมตริกซ์ทแยงมุม (The Diagonal Element) ที่สามารถคำนวณรากที่สองของค่าเจาะจง (Eigenvalue) ของ AA^T ได้ โดยค่าที่นำมาคำนวณ คือ สมาชิกของเมตริกซ์หลักที่ r ทำการคำนวณเพื่อหาเส้นสมมติแสดงปริมาณและทิศทางแนวตั้งฉากที่มีความสัมพันธ์กัน ทั้งนี้สมาชิกแนวเส้นทแยงมุมของเมตริกซ์ตัวที่ 1 ถึงตัวที่ r มีค่ามากกว่าศูนย์ $\sigma_1 \ge \sigma_2 \ldots \ge \sigma_r > 0$ และสมาชิกแนวเส้นทแยงมุมของเมตริกซ์ตัวที่ r+1 จนถึงสมาชิกของเมตริกซ์ตัวที่ n มีค่าเท่ากับศูนย์ $\sigma_{r+1} = \sigma_n = 0$ กล่าวได้ว่าเป็น ค่าแบบเดี่ยวของเมตริกซ์ A_k (Singular Value of A_k) สำหรับกระบวนการแยก(Decomposition) ของเมตริกซ์ นั่นเอง

เมตริกซ์ U คือเมตริกซ์เชิงตั้งฉาก (The Orthogonal Matrix) ที่สามารถคำนวณค่าเวกเตอร์เจาะจง (Eigenvector) ของ AA^T เรียกว่าเวกเตอร์คำ (Term Vector) เป็นเวกเตอร์เชิงเดี่ยวทางซ้ายของเมตริกซ์ A_k (Left Singular Vector of A_k)

เมตริกซ์ V คือเมตริกซ์เชิงตั้งฉาก (The Orthogonal Matrix) ที่สามารถคำนวณค่าเวกเตอร์เจาะจง (Eigenvector) ของ $A^T A$ เรียกว่าเวกเตอร์เอกสาร (Document Vector) เป็นเวกเตอร์เชิงเดี่ยวทางขวาของเมตริกซ์ A_k (Right Singular Vector of A_k)

กำหนดให้ R(A) และ N(A) เป็นสัญลักษณ์ใช้แทนอันดับ และช่องว่างที่ไม่สำคัญของเมตริกซ์ A ตามลำดับ ดังนั้น

- คุณสมบัติในการพิจารณาอันดับของเมตริกซ์ มีดังนี้ Rank (A) = r
  $N(A) \equiv span\{v_{r+1},\ldots,v_n\}$ และ $R(A) \equiv span\{u_1,\ldots,u_r\}$ เมื่อ $U = [u_1 u_2 \ldots u_m]$ และ $V = [v_1 v_2 \ldots v_m]$
- การแตกเมตริกซ์ A ออก มีรูปแบบดังนี้ $A_k = \sum_{i=1}^{k} u_i \cdot \sigma_i \cdot v_i^T$
- ค่ามาตรฐาน คือ $\| A \|_F^2 = \sigma_1^2 + \ldots + \sigma_r^2$ และ $\| A \|_2^2 = \sigma_1$

ทฤษฎีที่ 2 ซึ่งเป็นทฤษฎีของ Eckart และ Young

กำหนดให้ SVD (A) ดังแสดงในสมการที่ 2.8 และ $r = rank(A) \le p = \min(m,n)$ สามารถหาเมตริกซ์ A_k ได้ดังนี้

$$A_k = \sum_{i=1}^{k} u_i \cdot \sigma_i \cdot v_i^T \qquad (2.9)$$

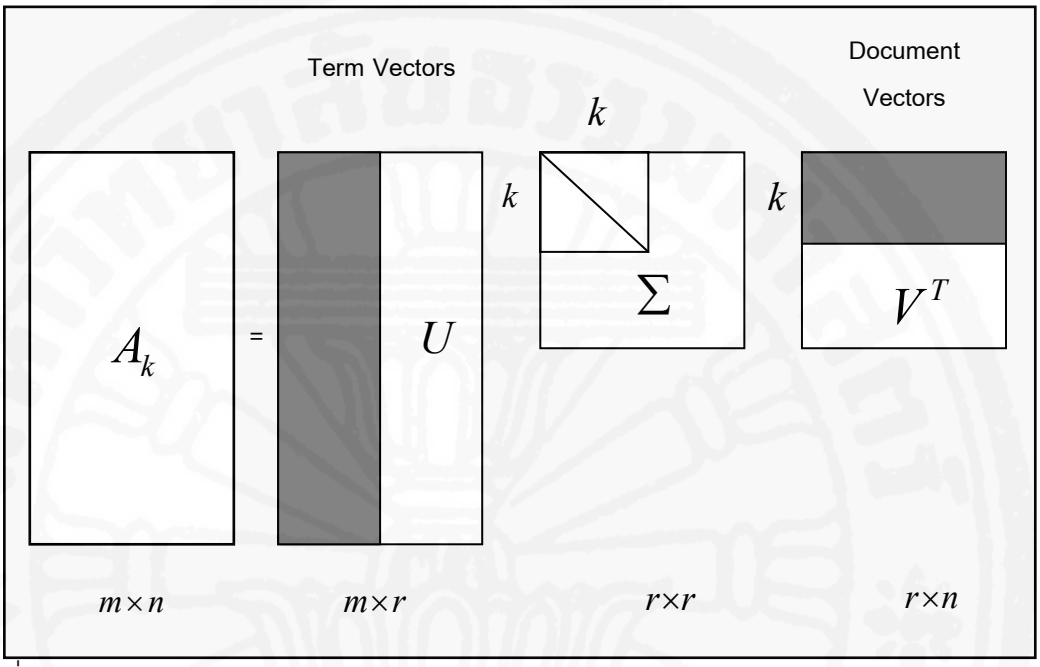
และ $$\min_{rank(B)=k} \| A - B \|_F^2 = \| A - A_k \|_F^2 = \sigma_{k+1}^2 + \ldots + \sigma_p^2$$

อาจกล่าวได้ว่า A_k เป็นเมตริกซ์ที่ถูกสร้างขึ้นมาจาก ค่า k ที่มากที่สุด ซึ่งก็คือหนึ่งในสามส่วนของเมตริกซ์ A ซึ่งค่า k ดังกล่าวเป็นอันดับที่ทำให้เมตริกซ์ A_k มีขนาดที่ใกล้เคียงกันกับเมตริกซ์ A มากที่สุด ในความเป็นจริงแล้ว A_k เป็นเมตริกซ์โดยประมาณที่ดีที่สุดของเมตริกซ์ A เพื่อที่จะพิจารณาค่าเปลี่ยนแปลงปกติใดๆ ของเมตริกซ์ดังนี้ $\min_{rank(B)=k} \| A - B \|_2 = \| A - A_k \|_2 = \sigma_{k+1}$ (Berry, Dumis and Brien, 1994)

จากทฤษฎีดังกล่าว สามารถสร้างเมตริกซ์ A_k ซึ่งเป็นเมตริกซ์ที่ได้จากการประมาณค่า k ดังแสดงในภาพที่ 2.8

ภาพที่ 2.8

แผนภาพอธิบายการสร้างเมตริกซ์ A_k จากการกระบวนการแยกค่าแบบเดี่ยว



ที่มา: Berry, Dumis and Brien (1994)

จากการศึกษาพบว่ากระบวนการแยกค่าแบบเดี่ยว (Singular Value Decomposition) มีขั้นตอนที่สำคัญนั่นก็คือ การเลือกขนาดของเมตริกซ์ หรือการประมาณค่า k โดยกำหนดให้ค่า $k \leq r$ เนื่องจากหากทำประมาณค่า k น้อยเกินไป จะไม่สามารถพิจารณาความสัมพันธ์ทางด้านความหมายของเนื้อหาที่สำคัญ (Semantic Space) ได้อย่างชัดเจน แต่ถ้าหากทำการประมาณค่า k มากเกินไป ก็จะไม่สามารถกำจัดเนื้อหาที่ไม่สำคัญ (Noise) ออกไปได้ดังนั้นการพิจารณาความสัมพันธ์ทางด้านความหมายก็จะไม่ชัดเจน (Deerwester, Dumais and Harshman, 1990)

การประมาณค่า k เพื่อนำมาสร้างเมตริกซ์ A_k นั้นจึงเป็นการเลือกค่า k ที่ดีและเหมาะสมที่สุด ค่า k ที่ได้จึงมาจากการทำการทดลองเพื่อหาค่า k สำหรับงานวิจัยชิ้นนั้นๆ จากการศึกษางานวิจัยที่เกี่ยวข้อง พบว่าการเลือกค่า k นี้เป็นปัญหาที่สำคัญ เนื่องจากไม่มีการกำหนดค่าที่แน่นอน ดังนั้นในแต่ละงานวิจัยจึงมีการเลือกค่า k ที่แตกต่างกันออกไป ยกตัวอย่างเช่น งานวิจัยเรื่อง การแนะนำเทคนิคการวิเคราะห์หาความหมายแอบแฝง (An Introduction to Latent Semantic Analysis) ของ Landauer Foltz และ Laham (1998) มีการกำหนดค่า k เท่ากับ 2 งานวิจัยเรื่อง การใช้พีชคณิตเชิงเส้นสำหรับศาสตร์การค้นคืนข้อมูลที่ชาญฉลาด (Linear Algebra

for Intelligent Information Retrieval) ของ Berry Dumais และ Brien (1994) มีการกำหนดค่า k เท่ากับ 2 งานวิจัยเรื่อง การย่อความภาษาไทยโดยการวิเคราะห์หาความหมายแอบแฝง (The Thai Text Summarization Using Latent Semantic Analysis) ของ อัษฎางค์ แตงไทย (2548) มีการกำหนดค่า k เท่ากับ 75 และงานวิจัยเรื่อง การประเมินคุณภาพการถอดความภาษาไทยด้วยการวิเคราะห์หาความหมายแอบแฝง (Assessment of Thai Paraphrases Using Latent Semantic Analysis) ของ ศุภวรรณ เปรมกุศลชัย (2548) มีการกำหนดค่า k สำหรับบทคัดย่อวิทยานิพนธ์ฉบับที่ 1 เท่ากับ 69 บทคัดย่อวิทยานิพนธ์ฉบับที่ 2 และ 3 เท่ากับ 109

จากตัวอย่างดังกล่าว สังเกตเห็นได้ว่าขนาดเมตริกซ์หรือค่า k นั้นมีการกำหนดค่าที่หลากหลายขึ้นอยู่กับความเหมาะสมสำหรับงานวิจัยนั้นๆ แต่อย่างไรก็ตามบางงานวิจัยก็ไม่สามารถที่จะทำการทดลองค่า k ทุกค่าได้ ทั้งนี้อาจจะเป็นเพราะว่างานวิจัยนั้นมีกลุ่มของข้อมูลที่หลากหลายและมีขนาดของข้อมูลที่ใหญ่มาก ทำให้ยากต่อการทำการทดลองค่า k ทุกค่า ด้วยเหตุนี้งานวิจัยดังกล่าวจะพิจารณาค่า k จากทฤษฎีการประมาณค่า k ที่ดีที่สุด (อัษฎางค์ แตงไทย, 2548) ซึ่งสามารถเลือกขนาดเมตริกซ์หรือค่า k ได้จากค่าผิดพลาด (Error) จากเมตริกซ์ A_k ไปยังเมตริกซ์ A ซึ่งเรียกค่าดังกล่าวนี้ว่า ค่าเปลี่ยนแปลงความสัมพันธ์ (Relative Change) (ศุภวรรณ เปรมกุศลชัย, 2548) ดังแสดงการคำนวณในสมการต่อไปนี้

$$\text{Relative Change} = \frac{\| A - A_k \|_F}{\| A \|_F} \qquad (2.10)$$

โดยที่ กำหนดให้กระบวนการแยกค่าแบบเดี่ยวของเมตริกซ์ A ดังแสดงในสมการที่ 2.1 และกำหนดให้ $Rank(A) = r \le p = \min(m,n)$

$$\| A - A_k \|_F = \min_{rank(B) \le k} \| A - B \|_F = \sqrt{\sigma_{k+1}^2 + \ldots + \sigma_{r_A}^2}$$

และ $$\| A \|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \ldots + \sigma_{r_A}^2}$$

(Berry, Dumais and Brien, 1994)

2.1.1.5 เทคนิคการวิเคราะห์หาความหมายแอบแฝง

การวิเคราะห์หาความหมายแอบแฝง (Latent Semantic Analysis: LSA) หรือ แอลเอสเอ เป็นเทคนิคหนึ่งในกระบวนการภาษาธรรมชาติ (Natural Language Processing: NLP)หรือ เอ็นแอลพี ที่มีลักษณะเฉพาะในการหาความหมายโดยใช้แบบจำลองเวกเตอร์ ซึ่งในการประยุกต์ใช้กับศาสตร์การค้นคืนข้อมูล อาจเรียกเทคนิคนี้ว่า ดัชนีแสดงความหมายแอบแฝง (Latent Semantic Indexing: LSI) หรือ แอลเอสไอ และอาจจะเรียกกระบวนในการค้นคืนคำที่ต้องการค้นหาจากเอกสารที่มีการใช้กระบวนการแยกค่าแบบเดี่ยว (Singular Value Decomposition: SVD) หรือ เอสวีดี ของคำโดยเมตริกซ์เอกสาร ดังกล่าวนี้ว่า ดัชนีแสดงความหมายแอบแฝงก็ได้เช่นเดียวกัน เนื่องจากว่ากระบวนการนี้สามารถทำการแสดงความสัมพันธ์ระหว่างคำที่มีความสำคัญกับเอกสารที่ต้องการค้นหาได้ ทั้งนี้เพราะว่าคำที่มีความสำคัญในแง่การเป็นตัวแทนของเอกสารนั้นจะไม่ปรากฏเด่นชัดให้หาได้โดยง่าย (Berry, Dumais and Brien, 1994) จึงจำเป็นที่จะต้องอาศัยกระบวนการดัชนีแสดงความหมายแอบแฝงและกระบวนการแยกค่าแบบเดี่ยว มาทำการค้นหาคำที่เป็นตัวแทนของเอกสารเพื่อวิเคราะห์หาความหมายแอบแฝง ต่อไปนั่นเอง

เทคนิคการวิเคราะห์หาความหมายแอบแฝงนั้น จะใช้กระบวนการแยกค่าแบบเดี่ยว กับเมตริกซ์ที่นำเข้าสู่ระบบ ดังตัวอย่างที่จะแสดงการวิเคราะห์ของเทคนิคการวิเคราะห์หาความหมายแอบแฝง ซึ่งตัวอย่างนี้ได้เลือกใช้ประโยคที่มีการบันทึกข้อความทางด้านเทคนิคทั้งหมด 9 ข้อความ โดยเป็นข้อความทางด้านการติดต่อสื่อสารระหว่างมนุษย์กับคอมพิวเตอร์ (Human Computer Interface: HCI) หรือ เฮชซีไอ จำนวน 5 ข้อความ และข้อความทางด้านทฤษฎีกราฟที่เกี่ยวกับคณิตศาสตร์ (Mathematical Graph Theory) จำนวน 4 ข้อความ ซึ่งจะสังเกตเห็นได้ว่าทั้งสองหัวข้อไม่มีความเกี่ยวข้องกันทางด้านหลักการ ดังแสดงรายละเอียดของข้อความในตารางที่ 2.6

ตารางที่ 2.6

แสดงข้อความบันทึกความรู้ทางด้านเทคนิค

ตัวอย่างข้อมูล: ข้อความเกี่ยวกับบันทึกความรู้ทางด้านเทคนิค
c1 : Human machine interface for ABC computer applications
c2 : A survey of user opinion of computer system response time
c3 : The EPS user interface management system
c4 : System and human system engineering testing of EPS
c5 : Relation of user perceived response time to error measurement
m1: The generation of random, binary, ordered trees
m2: The intersection graph of paths in trees
m3: Graph minors IV: Widths of trees and well-quasi-ordering
m4: Graph minors: A survey

ที่มา: Landauer Foltz and Laham (1998)

ข้อความทั้งหมดคือจำนวนหลักของเมตริกซ์เริ่มต้นมีค่าเท่ากับ 9 และจากการนับจำนวนคำที่ปรากฏขึ้นในข้อความอย่างน้อย 2 ข้อความ พบว่ามีคำที่มีลักษณะดังกล่าว ทั้งหมด 12 คำ ดังแสดงด้วยตัวเอียงในตารางที่ 2.6 ดังนั้น คำที่ปรากฏในข้อความอย่างน้อย 2 ข้อความนั้นคือจำนวนแถวของเมตริกซ์เริ่มต้นมีค่าเท่ากับ 12 ดังนั้นเมตริกซ์เริ่มต้นจะมีขนาดเมตริกซ์ 9 x 12 และกำหนดให้เมตริกซ์เริ่มต้น คือ เมตริกซ์ X

โดยค่าความถี่ของคำ (แถว) ที่ปรากฏในข้อความ (หลัก) อย่างน้อย 2 ข้อความ คือ ค่าใน แต่ละเซลล์ของเมตริกซ์ ดังแสดงในภาพที่ 2.9

ภาพที่ 2.9

แสดงเมตริกซ์ของคำที่ปรากฏในข้อความ

$\{A\}$ =

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

r (human.user) = - 0.38

r (human.minors) = - 0.29

ที่มา: Landauer Foltz และ Laham (1998)

จากนั้นระบบก็จะดำเนินการแยกค่าเชิงเส้น (Linear Decomposition) ผ่านกระบวนการแยกค่าแบบเดี่ยว (Singular Value Decomposition) แล้วให้ผลลัพธ์เป็นเมตริกซ์ต่างๆ ดังแสดงในภาพที่ 2.10

ภาพที่ 2.10

แสดงเมตริกซ์ผลลัพธ์ที่ได้จากการดำเนินการผ่านกระบวนการแยกค่าแบบเดี่ยว

$$\{A\} = \{U\}\{\Sigma\}\{V\}^T$$

$\{U\} =$

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

ที่มา : Deerwester Dumais and Harshman (1986)

ภาพที่ 2.10 (ต่อ)

แสดงเมตริกซ์ผลลัพธ์ที่ได้จากการดำเนินการผ่านกระบวนการแยกค่าแบบเดี่ยว

$\{\Sigma\} =$

3.34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	2.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	2.35	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	1.64	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	1.50	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	1.31	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.85	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.56	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.36

ที่มา : Deerwester Dumais and Harshman (1986)

$\{V\} =$

0.20	-0.06	0.11	-0.95	0.05	-0.08	0.18	-0.01	-0.06
0.61	0.17	-0.50	-0.03	-0.21	-0.26	-0.43	0.05	0.24
0.46	-0.13	0.21	0.04	0.38	0.72	-0.24	0.01	0.02
0.54	-0.23	0.57	0.27	-0.21	-0.37	0.26	-0.02	-0.08
0.28	0.11	-0.51	0.15	0.33	0.03	0.67	-0.06	-0.26
0.00	0.19	0.10	0.02	0.39	-0.30	-0.34	0.45	-0.62
0.01	0.44	0.19	0.02	0.35	-0.21	-0.15	-0.76	0.02
0.02	0.62	0.25	0.01	0.15	0.00	0.25	0.45	0.52
0.08	0.53	0.08	-0.03	-0.60	0.36	0.04	-0.07	-0.45

ที่มา : Deerwester Dumais and Harshman (1986)

ภาพที่ 2.10 (ต่อ)

แสดงเมตริกซ์ผลลัพธ์ที่ได้จากการดำเนินการผ่านกระบวนการแยกค่าแบบเดี่ยว

$\{V\}^T =$

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

ที่มา : Deerwester Dumais and Harshman (1986)

จากการดำเนินการแยกค่าเชิงเส้น (Linear Decomposition) แล้วจะได้ผลลัพธ์คือ เมตริกซ์ U เมตริกซ์ $\sum$ และเมตริกซ์ V ทั้งนี้สามารถสร้างเมตริกซ์เริ่มต้น คือเมตริกซ์ A ได้ดังสมการนี้

$$\{A\} = \{U\}\{\Sigma\}\{V\}^T \qquad (2.11)$$

โดยที่กระบวนการแยกค่าแบบเดี่ยวจะให้ค่าเมตริกซ์ V เท่านั้น หากต้องการสร้างเมตริกซ์ A จะต้องทำการสับเปลี่ยนตำแหน่งตามหลักการที่เรียกว่าการ Transpose แล้วจึงนำเมตริกซ์ V^T ที่ได้ไปทำการคำนวณเพื่อหาค่าเมตริกซ์ A ตามสูตรดังกล่าวข้างต้น

นอกจากนี้แล้วเมตริกซ์ A ที่ได้มานั้นยังไม่สามารถสร้างโครงสร้างความหมายหรือความสัมพันธ์ของเนื้อหาที่ชัดเจนได้ จึงจำเป็นต้องเลือกขนาดเมตริกซ์ หรือค่า k ซึ่งจากการศึกษาพบว่าค่า k ที่ Deerwester Dumais และ Harshman (1986) เลือกใช้ในงานวิจัยคือ k = 2 ดังจะ

สังเกตเห็นได้จากการแรเงาสีในภาพที่ 2.10 เพื่อนำมาสร้างเป็นเมตริกซ์ $\hat{A}$ โดย $\hat{A} = U_k \sum_k V_k^T$ และสามารถคำนวณเพื่อสร้างเมตริกซ์ A_k ได้ดังสูตร

$$\{A_k\} = \{U_k\}\{\Sigma_k\}\{V_k\}^T \qquad (2.12)$$

และแสดงตัวอย่างการคำนวณได้ดังภาพที่ 2.11

ภาพที่ 2.11
แสดงเมตริกซ์ที่ได้จากการดำเนินการเลือกขนาดเมตริกซ์หรือค่า k = 2

$\{U_k\}$ =

0.22	-0.11
0.20	-0.07
0.24	0.04
0.40	0.06
0.64	-0.17
0.27	0.11
0.27	0.11
0.30	-0.14
0.21	0.27
0.01	0.49
0.04	0.62
0.03	0.45

$\{\Sigma_k\}$ =

3.34	0.00
0.00	2.54

$\{V_k\}^T$ =

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53

ที่มา : Deerwester Dumais and Harshman (1986)

จากที่ได้ผลลัพธ์เป็นเมตริกซ์ $\{U_k\}\{\Sigma_k\}\{V_k\}^T$ แล้ว สามารถสร้างเมตริกซ์ $\{A_k\}$ ได้ดังแสดงในภาพที่ 2.12

ภาพที่ 2.12

แสดงการสร้างเมตริกซ์ A_k และผลลัพธ์

$$A_k = U_k \sum_k V_k^T$$

$\{A_k\} =$

0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
0.14	0.37	0.33	0.40	0.17	-0.03	-0.07	-0.10	-0.04
0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.43
-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

ที่มา : Deerwester Dumais and Harshman (1986)

2.1.2 ทฤษฎีในศาสตร์ปัญญาประดิษฐ์

ในศาสตร์ปัญญาประดิษฐ์ ซึ่งเป็น “วิชาที่ว่าด้วยการศึกษาเพื่อให้เข้าใจถึงความฉลาดและสร้างระบบคอมพิวเตอร์ที่ชาญฉลาด และนำมาทำงานแทนหรือช่วยมนุษย์ทำงานที่ต้องใช้ความฉลาดนั้นๆ” (บุญเสริม กิจศิริกุล, 2546) และจากการศึกษาเทคนิคการเรียนรู้ของเครื่อง (Machine Learning) ในส่วนของโครงข่ายประสาทเทียม (Artificial Neural Networks) พบว่า มีการให้คำจำกัดความไว้ว่าเป็น “ระบบการคำนวณชนิดหนึ่งซึ่งสร้างขึ้นจากหน่วยประมวลผลอย่างง่ายจำนวนมาก ที่เชื่อมต่อเข้าด้วยกันอย่างหนาแน่น และประมวลผลสารสนเทศโดยการตอบสนองต่อข้อมูลจากภายนอก ในสถานะที่ไม่คงตัว” (สารานุกรมไทยสำหรับเยาวชนฉบับกาญจนาภิเษก เล่ม 25, โครงการสารานุกรมไทยสำหรับเยาวชน, 2544)

2.1.2.1 เทคนิคโครงข่ายประสาทเทียม

แนวคิดในการสร้างโครงข่ายประสาทเทียมนั้นเกิดขึ้นจากการที่มนุษย์เรามีความคิดที่จะจำลองการทำงานของระบบสมองของมนุษย์เพื่อให้ทำงานที่เกี่ยวข้องกับความฉลาด (Intelligence) ได้อย่างมีประสิทธิภาพเทียบเท่ากับมนุษย์ อาทิเช่น การเรียนรู้และเข้าใจจากประสบการณ์ที่ผ่านมา การตอบสนองต่อข้อมูลที่คลุมเครือหรือขัดแย้งกัน ความสามารถในการให้เหตุผลเพื่อแก้ปัญหาได้อย่างมีประสิทธิภาพ ความสามารถในการจัดการและแก้ไขสถานการณ์ที่ซับซ้อนได้ ความสามารถที่จะหาความรู้และใช้ความรู้นั้นได้ และความสามารถในการที่จะคิดโดยใช้เหตุผล เป็นต้น (บุญเสริม กิจศิริกุล, 2546) เนื่องจากว่าความสามารถในการทำงานเหล่านี้ระบบสมองของมนุษย์มีความสามารถที่จะแก้ปัญหาได้อย่างมีประสิทธิภาพสูงกว่าคอมพิวเตอร์มาก

ดังนั้นการพัฒนาคอมพิวเตอร์ไปสู่ระบบการประมวลผลรูปแบบใหม่ที่สามารถเรียนรู้และคิดได้ จึงเป็นแนวทางหนึ่งที่นักวิทยาศาสตร์ให้ความสนใจเป็นอย่างมาก จึงพยายามศึกษาการสร้าง แบบจำลองโครงข่ายประสาทเทียม โดยการศึกษาทำความเข้าใจกระบวนการทำงานของสมองของสิ่งมีชีวิต และอธิบายกระบวนการทำงานนั้นด้วยแบบจำลองเชิงคณิตศาสตร์ จากนั้นจึงออกแบบสถาปัตยกรรมโครงข่ายประสาทเทียมทั้งนี้ก็เพื่อให้สามารถออกแบบระบบคอมพิวเตอร์ และสามารถเขียนโปรแกรมคอมพิวเตอร์ที่จะทำงานตามแบบจำลองเชิงคณิตศาสตร์ได้ (ปริชัย วิสุทธิ์เมธากุล, 2548) แนวทางการพัฒนาคอมพิวเตอร์เพื่อสร้างแบบจำลองโครงข่ายประสาทเทียมนั้นมีขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 ทำการศึกษาและทำความเข้าใจกระบวนการทำงานของสมองของสิ่งมีชีวิต

เซลล์ประสาทของสิ่งมีชีวิต (Neuron) หรือ นิวรอน มีองค์ประกอบพื้นฐานที่สำคัญ 3 ส่วนดังต่อไปนี้

- โซมา (Soma) หรือ เซลล์บอดี้ (Cell Body) คือตัวเซลล์ทำหน้าที่รวบรวมข้อมูลที่ได้จากปลายเซลล์ประสาท เปรียบเสมือนตัวรวบรวมค่าน้ำหนักของสัญญาณไฟฟ้า
- เดนไดร์ท (Dendrite) คือเส้นใยบางๆ ที่ทำหน้าที่ในการรับสัญญาณไฟฟ้าเข้าสู่เซลล์ประสาท เปรียบเสมือน อินพุต ที่เป็นตัวรับข้อมูลเข้ามา
- แอ็กซอน (Axon) คือเส้นใยบางๆ ที่ทำหน้าที่ในการส่งผ่านสัญญาณไฟฟ้าไปยังเซลล์ประสาทอื่นๆ เปรียบเสมือน เอาต์พุต ที่ส่งผลลัพธ์หรือคำตอบไปยังเซลล์ประสาทอื่นต่อไป

ในส่วนบริเวณที่เป็นรอยต่อของเซลล์ประสาทของสิ่งมีชีวิตที่เชื่อมต่อระหว่างปลายของแอ็กซอนกับเดนไดร์ท จะถูกเรียกกว่า ไซแนปส์ (Synapse)

คุณลักษณะสำคัญของ ไซแนปส์ ก็คือ แรงกระตุ้นของสัญญาณที่ถูกส่งผ่านจะขึ้นอยู่กับการเชื่อมต่อที่เหนียวแน่น กล่าวคือ สัญญาณที่ถูกส่งผ่านไซแนปส์ อาจจะมีสภาพเป็นสัญญาณกระตุ้น หรือสัญญาณยับยั้ง ก็ได้ ทั้งนี้ขึ้นอยู่กับชนิดของสัญญาณเคมีที่เคลื่อนที่ผ่านไซแนปส์ เนื่องจากแต่ละไซแนปส์ทำหน้าที่เป็นตัวปรับเปลี่ยนสัญญาณไฟฟ้าที่ส่งมาจากเซลล์ประสาทตัวอื่นๆ ดังนั้นก่อนที่จะส่งสัญญาณผ่าน เดนไดร์ท เข้าสู่เซลล์บอดี้ จะมีการปรับสัญญาณดังกล่าวโดยขึ้นอยู่กับความเหนียวแน่นในการเชื่อมต่อกันระหว่างรอยต่อของไซแนปส์

กล่าวคือ ความเหนียวแน่นนี้ขึ้นอยู่กับการเรียนรู้ของเซลล์สมอง ในที่นี้จะเปรียบเทียบความเหนียวแน่นด้วยค่าน้ำหนัก หากความเหนียวแน่นในการเชื่อมต่อกันสูงหรือค่าน้ำหนักมาก สัญญาณไฟฟ้าก็จะสามารถส่งผ่านถึงกันได้มาก แต่หากความเหนียวแน่นในการเชื่อมต่อกันต่ำหรือค่าน้ำหนักน้อย สัญญาณไฟฟ้าจะส่งผ่านรอยต่อได้น้อย นอกจากนั้นยังสามารถพิจารณาประเภทของสัญญาณได้ว่าเป็นสัญญาณกระตุ้นหากค่าน้ำหนักที่ได้มีค่าเป็นบวก หรือเป็นสัญญาณยับยั้งหากค่าน้ำหนักที่ได้มีค่าเป็นลบ

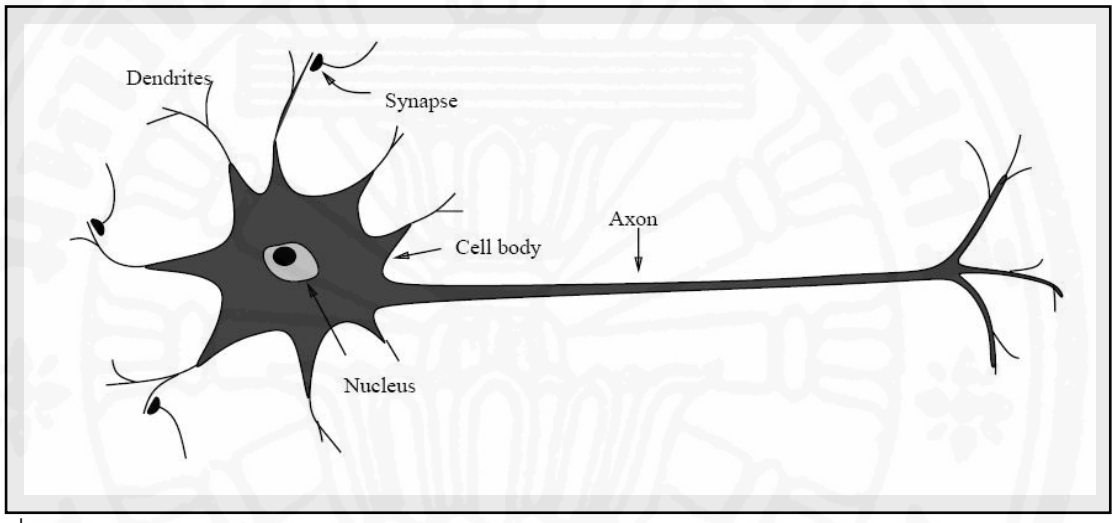
กระบวนการทำงานของเซลล์ประสาทของสิ่งมีชีวิต เริ่มต้นจากการที่สัญญาณไฟฟ้าที่ถูกส่งมาถึงปลายแอ็กซอนจะถูกกระตุ้นให้เกิดการส่งสัญญาณเคมีผ่านไซแนปส์ และสัญญาณเคมี

ดังกล่าวจะไปถึงเดนไดร์ท จากนั้นเดนไดร์ทจะทำการตีความเป็นสัญญาณไฟฟ้าอีกครั้ง เพื่อวิ่งเข้าสู่เซลล์ประสาทต่อไป

โดยจะแสดงลักษณะของโครงข่ายเซลล์ประสาทภายในสมองของมนุษย์ ดังภาพที่ 2.13

ภาพที่ 2.13

แสดงโครงข่ายเซลล์ประสาทในสมองมนุษย์



ที่มา: Jain, Mao และ Mohiuddin (1996)

ขั้นตอนที่ 2 อธิบายหลักการทำงานดังกล่าวด้วยแบบจำลองทางคณิตศาสตร์ เพื่อออกแบบสถาปัตยกรรมโครงข่ายประสาทเทียม

โดยโครงข่ายประสาทเทียมนั้นมีการเชื่อมต่อระหว่างองค์ประกอบภายในลักษณะคล้ายเส้นใยประสาท หรือที่เรียกกันโดยทั่วไปว่า เซลล์ประสาท หรือ นิวรอน โดยที่เซลล์ประสาทนี้ถือได้ว่าเป็นองค์ประกอบพื้นฐานที่สำคัญในการทำงาน ทั้งในการรับค่าและส่งค่าออกไปยังเซลล์ประสาทอื่นๆ เพื่อทำงานเป็นระบบร่วมกันอย่างมีประสิทธิภาพ แต่ถึงกระนั้น องค์ประกอบพื้นฐานของโครงข่ายประสาทเทียมมิได้มีอยู่แค่ เซลล์ประสาท เท่านั้น หากแต่ประกอบไปด้วย เพอร์เซปตรอน (Perceptron) ไบโพลาร์ (Bipolar) และซิกมอยด์ (Sigmoid Unit) เป็นต้น และทุกองค์ประกอบพื้นฐานนั้นก็มีหน้าที่การทำงานที่แตกต่างกัน ดังจะอธิบายในรายละเอียดต่อไป

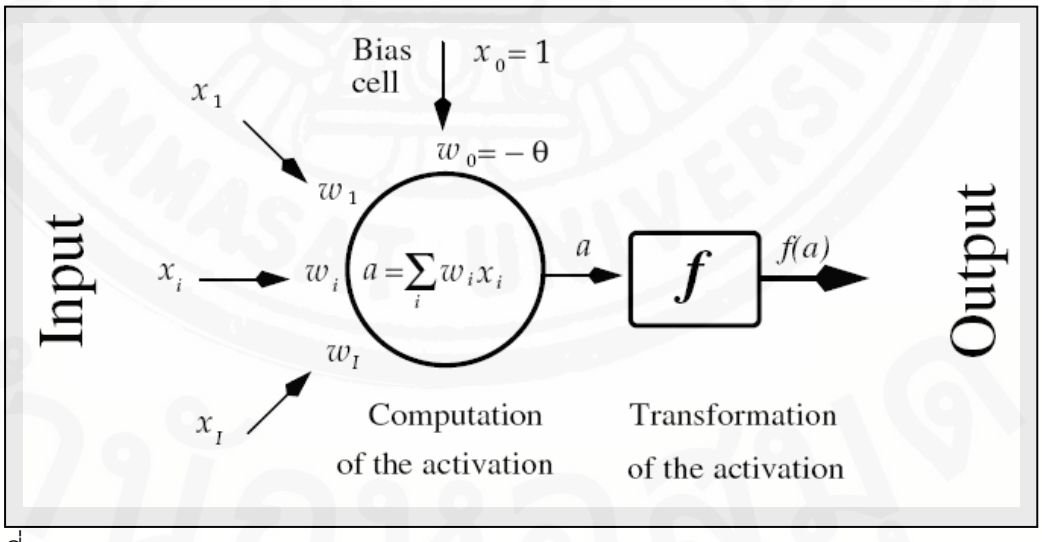
เซลล์ประสาท เป็นหน่วยหนึ่งในโครงข่ายประสาทเทียม ที่ทำหน้าที่รวบรวมข้อมูลจากหน่วยอื่นๆ หรือจากข้อมูลนำเข้า หรือเรียกกันโดยทั่วไปว่า อินพุต จากภายนอก เข้าสู่ระบบซึ่ง

เชื่อมโยงกันด้วยค่าน้ำหนัก (Weighted Connections) ซึ่งเรียกว่า ไซน์เนปส์ และ ค่าน้ำหนักนี้ก็จะถูกเรียกว่า ค่าน้ำหนักไซน์เนปส์ (Synaptic Weight)

ในแต่ละหน่วยจะแสดงถึง รูปแบบของเซลล์ประสาท และการเปลี่ยนแปลงของอินพุตจนกระทั่งถึงข้อมูลนำออก หรือที่เรียกกันโดยทั่วไปว่า เอาต์พุต ซึ่งการเปลี่ยนแปลงที่จะนำเสนอนี้เริ่มต้นจากการดำเนินการกระตุ้นให้เซลล์ประสาททำการคำนวณค่าน้ำหนักรวมให้กับอินพุต จากนั้นจึง ทำการกระตุ้นให้มีการเปลี่ยนแปลงค่าน้ำหนักโดย ฟังก์ชันการเปลี่ยนแปลง (Transfer Function)

โดยกำหนดรูปแบบของตัวแปรดังต่อไปนี้ อินพุต แทนด้วย x_i น้ำหนักแต่ละค่า แทนด้วย w_i ค่าการกระตุ้นคำนวณได้จาก $a = \sum x_i w_i$ และ เอาต์พุต แทนด้วย o สามารถคำนวณค่าเอาต์พุตได้จาก $o = f(a)$ และตัวอย่างที่มีค่าของอินพุตเป็นจำนวนจริง (Real Number) สามารถใช้ ฟังก์ชันการเปลี่ยนแปลง (Transfer Function) ได้ ดังแสดงหน่วยพื้นฐานของเซลล์ประสาท (The Basic Neural Unit) ในภาพที่ 2.14

ภาพที่ 2.14
แสดงหน่วยพื้นฐานของเซลประสาท



ที่มา : Herve Abdei (2003)

เพอร์เซปตรอน (Perceptron) มีการเรียนรู้แบบมีผู้สอน (Supervised Learning) โดยเพอร์เซปตรอนจะถูกสอนว่าข้อมูลตัวอย่างที่สอนเข้าไปแต่ละแบบนั้นจัดเป็นข้อมูลชนิดใดบ้าง

หากปัญหาและข้อมูลตัวอย่างมีความเหมาะสมกัน เพอร์เซปตรอนจะสามารถระบุชนิดของข้อมูลที่ไม่เคยเห็นมาก่อนได้อย่างถูกต้อง ดังนั้น เพอร์เซปตรอนจึงมีความเหมาะสมกับงานการระบุชนิดหรือการแบ่งประเภทข้อมูล

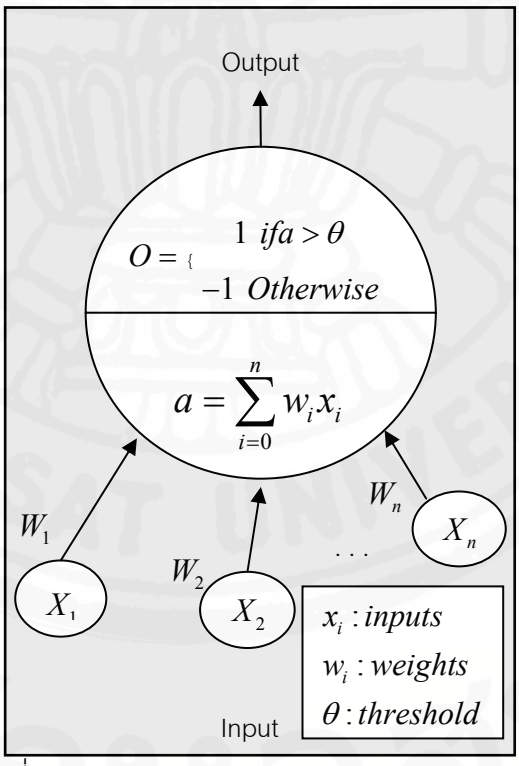
ยกตัวอย่างเช่น เพอร์เซปตรอนรับค่าเวกเตอร์ (Vector) จำนวนจริงเข้ามาเป็นอินพุต และทำการคำนวณผลรวมเชิงเส้น (Linear Combination) ของอินพุตเหล่านั้น จากนั้น

- ระบบจะให้ค่าเอาต์พุต เป็น 1 ถ้าผลรวมเชิงเส้นมีค่ามากกว่าค่าที่กำหนด
- ระบบจะให้ค่าเอาต์พุต เป็น -1 ถ้าหากผลรวมเชิงเส้นน้อยกว่าหรือเท่ากับค่าที่กำหนด

ดังแสดงในภาพที่ 2.15

ภาพที่ 2.15

แสดงการทำงานของเพอร์เซปตรอน



ที่มา : David Kriegman (2001)

หากพิจารณาเอาต์พุตที่ได้จาก ระนาบเพอร์เซปตรอน จะพบว่า

- ข้อมูลเอาต์พุตเป็น 1 สำหรับตัวอย่างที่อยู่บนด้านหนึ่ง
- ข้อมูลเอาต์พุตเป็น -1 สำหรับตัวอย่างที่อยู่อีกด้านหนึ่งหรือด้านตรงข้าม

ดังนั้นจึงทำให้เซตของตัวอย่างบางตัวอย่างนั้นสามารถแยกออกมาได้เป็นตัวอย่างบวกและตัวอย่างลบ โดยระนาบที่เกิดขึ้นได้ ดังจะเรียกเซตของตัวอย่างที่สามารถแบ่งในลักษณะดังกล่าวนี้ได้ว่า เซตตัวอย่างที่สามารถแยกได้แบบเชิงเส้น (Linearly Separable Sets of Examples) แต่ก็มีบางตัวอย่างที่ ไม่สามารถแทนด้วยเพอร์เซปตรอนเดี่ยวได้ จำเป็นต้องอาศัยเพอร์เซปตรอนมาต่อกันหลายชั้นเพื่อให้สามารถจำแนกตัวอย่างเหล่านั้นได้ ดังนั้นจึงมีการสร้างโครงข่ายประสาทเทียมแบบหลายชั้น (Multilayer Neural Network)

สถาปัตยกรรมของโครงข่ายประสาทเทียม นั่นก็คือ การเชื่อมโยงเซลล์ประสาทเทียมจำนวนหนึ่งเข้าด้วยกันเป็นโครงข่ายประสาทเทียม ซึ่งการเชื่อมโยงนั้นสามารถทำได้หลากหลายรูปแบบ แต่ในทางปฏิบัติแล้วเทคนิคการเรียนรู้ของโครงข่ายประสาทเทียมมักจะถูกออกแบบมาเพื่อให้ใช้งานได้กับสถาปัตยกรรมโครงข่ายประสาทเทียมที่มีลักษณะเฉพาะเท่านั้น โดยลักษณะที่พบโดยทั่วไป ก็คือ การจัดเซลล์ประสาทเทียมเป็นชั้น (Layer) โดยชั้นที่รับข้อมูลเข้า เรียกว่า ชั้นอินพุต (Input Layer) ชั้นที่ทำการประมวลผลภายใน เรียกว่า ชั้นซ่อน (Hidden Layer) โดยโครงข่ายประสาทเทียมอาจจะมีชั้นซ่อนหลายชั้นได้ และสุดท้ายชั้นที่ให้ข้อมูลออก หรือให้ผลลัพธ์กับโครงข่าย เรียกว่า ชั้นเอาต์พุต (Output Layer) และโครงสร้างพื้นฐานจะมีการประกอบกันในรูปแบบป้อนไปข้างหน้า (Feedforward) ซึ่งสามารถแบ่งเป็น 2 แบบคือ โครงข่ายประสาทเทียมแบบชั้นเดียว และ โครงข่ายประสาทเทียมแบบหลายชั้น

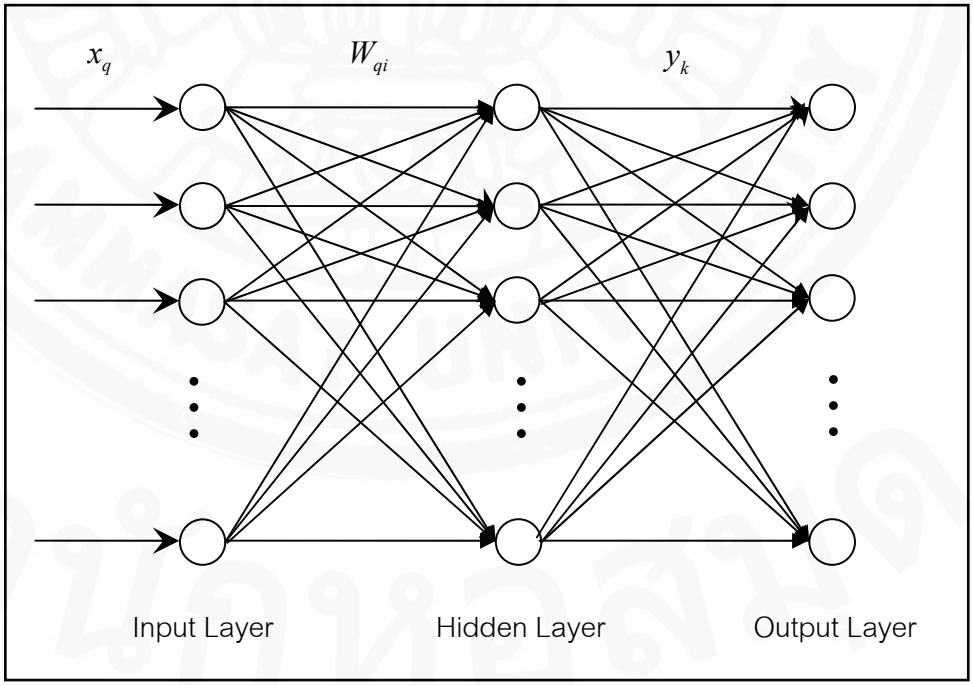
ยกตัวอย่างประเภทของโครงข่ายประสาทเทียมแบบหลายชั้นที่เป็นที่นิยมใช้ในงานวิจัยโดยทั่วไป นั่นก็คือ โครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับหลายชั้น (Multilayer Backpropagation Neural Network) ซึ่งเป็นเทคนิคหนึ่งที่มีความสามารถในการสร้างพื้นผิวการตัดสินใจแบบไม่เป็นเชิงเส้น (Nonlinear Decision Surface) ซึ่งสามารถแบ่งแยกตัวอย่างได้ดีกว่าการสร้างพื้นผิวการตัดสินใจแบบเชิงเส้น (Linear Decision Surface) (สุกรี สินธุภิญโญ, 2544) ทั้งนี้จะกล่าวถึงรายละเอียดการทำงานในหัวข้อต่อไป

2.1.2.2 **อัลกอริทึมการแพร่กระจายย้อนกลับ**

อัลกอริทึมการแพร่กระจายย้อนกลับ คือ อัลกอริทึมการเรียนรู้ค่าเวกเตอร์น้ำหนักสำหรับข่ายงานแบบป้อนไปข้างหน้าหลายชั้น (Multilayer Feedforward Network) โดยหาค่าผิดพลาดต่ำสุดระหว่างเอาต์พุตของโครงข่าย กับเอาต์พุตเป้าหมาย

โครงข่ายแบบป้อนไปข้างหน้าแบบหลายชั้น ประกอบไปด้วย ชั้นอินพุต ชั้นซ่อน และชั้นเอาต์พุต ซึ่งแต่ละบัพจะมีการเชื่อมต่อกันทั้งหมด (Fully Connection) โดยการเชื่อมต่อนั้นจะทำการเชื่อมต่อชั้นต่อชั้น คือ ชั้นอินพุต เชื่อมต่อไปยัง ชั้นซ่อน ชั้นซ่อน เชื่อมต่อไปยังชั้นเอาต์พุต โครงข่ายป้อนไปข้างหน้าแบบหลายชั้นนี้จะไม่ทำการเชื่อมย้อนกลับ กล่าวคือจะมีเส้นเชื่อมไปข้างหน้าเพียงอย่างเดียวเท่านั้น ไม่มีเส้นเชื่อมจากชั้นเอาต์พุตย้อนกลับมายังชั้นซ่อนหรือชั้นอินพุต ตัวอย่างแสดงข่ายงานแบบป้อนไปข้างหน้าหลายชั้น ดังแสดงในภาพที่ 2.16

ภาพที่ 2.16
แสดงโครงสร้างของข่ายงานแบบป้อนไปข้างหน้าหลายชั้น



การพิจารณาเลือกใช้งานโครงข่ายประสาทเทียมจำเป็นที่จะต้องศึกษาถึงโครงสร้างการทำงานและองค์ประกอบของโครงข่ายประสาทเทียมแบบที่เลือกนำมาประยุกต์ใช้

งานวิจัยชิ้นนี้นำเสนอโครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับหลายชั้น (Mulitlayer Backpropagation Neural Network) เป็นรูปแบบหนึ่งของโครงข่ายประสาทเทียมที่มีลักษณะโครงสร้างการทำงานเป็นข่ายงานแบบป้อนไปข้างหน้าหลายชั้น และมีความสามารถในการสร้างระนาบการตัดสินใจแบบไม่เป็นเชิงเส้น

โดยโครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับ มีความแตกต่างจาก เพอเซปตรอน ในส่วนขององค์ประกอบย่อยของโครงข่ายประสาทเทียม กล่าวคือ โครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับ สามารถทำงานกับฟังก์ชันไม่เชิงเส้น (Nonlinear Function) โดยใช้องค์ประกอบซิกมอยด์ (Sigmoid Unit)

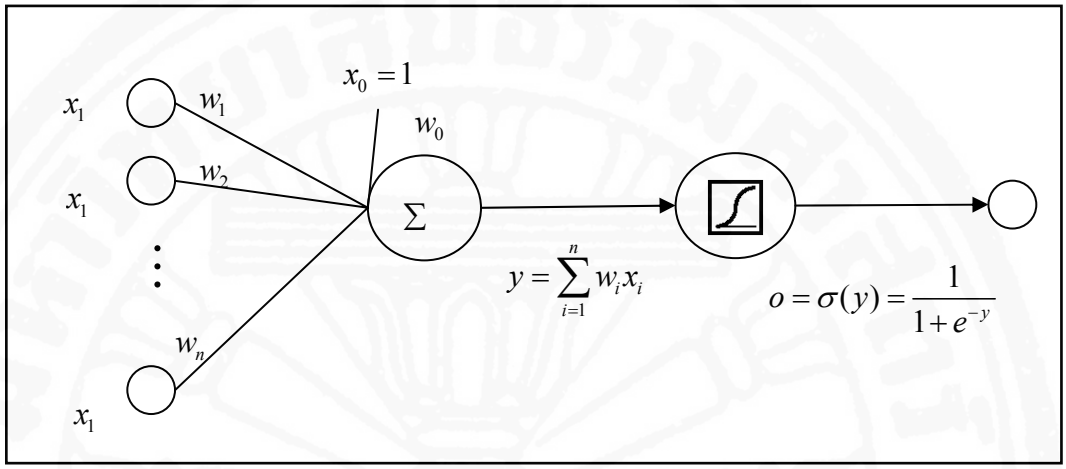
ลักษณะการทำงานขององค์ประกอบซิกมอยด์ จะดำเนินการโดยหาผลคูณของเวกเตอร์อินพุตกับค่าน้ำหนักของเส้นเชื่อมระหว่างอินพุตเซลล์ประสาทกับเซลล์ประสาทในชั้นถัดไป จากนั้นนำผลคูณที่ได้ไปผ่านตัวกรอง (Threshold) จนกระทั่งได้เอาต์พุตขององค์ประกอบซิกมอยด์ออกมาเป็นฟังก์ชันต่อเนื่อง (Continueous Function) ที่มีค่าเข้าใกล้ 0 หรือ 1 ในขณะที่เพอเซปตรอน (Perceptron) จะให้ค่าเอาต์พุตเป็นค่าไม่ต่อเนื่อง (สุกรี สินธุภิญโญ, 2544) ทั้งนี้เนื่องจากเพอเซปตรอนสามารถเรียนรู้ฟังก์ชันแยกเชิงเส้น (Linear Function) ได้เท่านั้น (บุญเสริม กิจศิริกุล, 2546)

การคำนวณหากฎการเรียนรู้สำหรับโครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับหลายชั้น ต้องใช้ฟังก์ชันกระตุ้นที่หาอนุพันธ์ได้ จากกฎเดลต้า (Delta Rule) (บุญเสริม กิจศิริกุล, 2546) เป็นกฎการเรียนรู้สำหรับหาค่าเวกเตอร์น้ำหนักของเพอร์เซปตรอนอีกกฎหนึ่ง ที่มีข้อดีคือการเรียนรู้จะลู่เข้าสู่ระนาบหลายมิติที่ให้ค่าผิดพลาดน้อยที่สุดแม้ว่าตัวอย่างนั้นจะเป็นฟังก์ชันแยกเชิงเส้นไม่ได้ กฎนี้ใช้หลักการ การเคลื่อนลงตามความชัน (Gradient Descent) เพื่อหาคำตอบ ซึ่งกฎนี้เป็นพื้นฐานของอัลกอริทึมแบบการแพร่กระจายย้อนกลับ จำเป็นต้องใช้ฟังก์ชันกระตุ้นที่หาอนุพันธ์ได้ ซึ่งก็คือ ฟังก์ชันเชิงเส้น(Linear Transfer Function) หรือ ฟังก์ชันซิกมอยด์ (Sigmoid Transfer Function) (บุญเสริม กิจศิริกุล, 2546) โดยองค์ประกอบและกระบวนการทำงานของแต่ละฟังก์ชันดังแสดงในภาพที่ 2.17, 2.18, 2.19 และ 2.20 ตามลำดับ

ภาพที่ 2.17

แสดงองค์ประกอบฟังก์ชันซิกมอยด์ (Sigmoid Transfer Function)

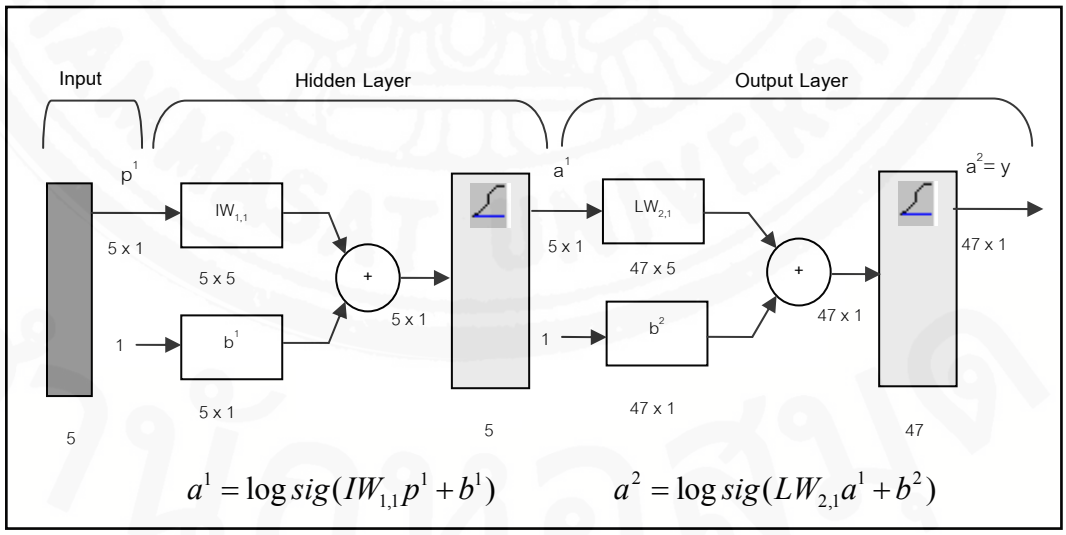
แบบลอกซิก (Log Sigmoid Transfer Function : logsig)



ภาพที่ 2.18

แสดงกระบวนการทำงานของโครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับหลายชั้น

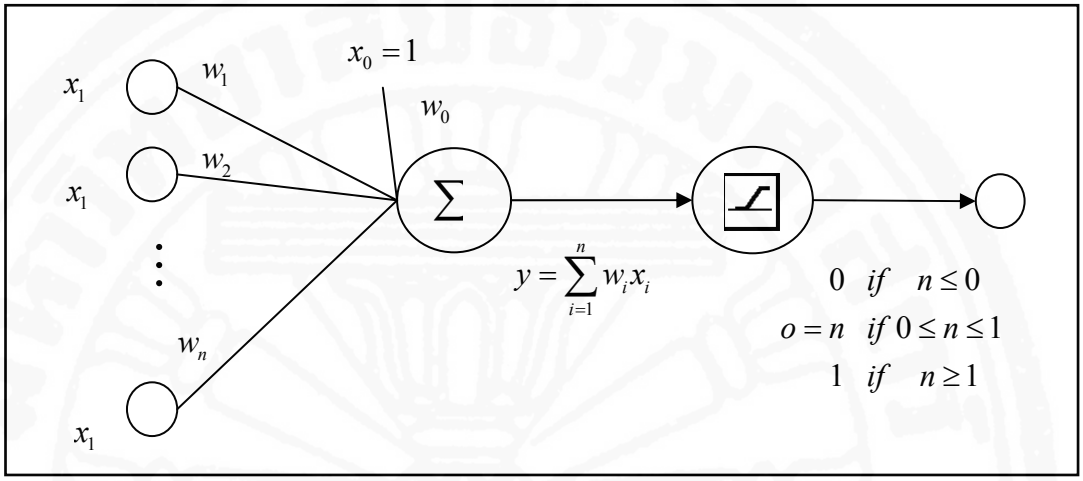
มีฟังก์ชันกระตุ้นแบบซิกมอยด์สำหรับงานวิจัยชิ้นนี้



ภาพที่ 2.19

แสดงองค์ประกอบฟังก์ชันเชิงเส้น(Linear Transfer Function)

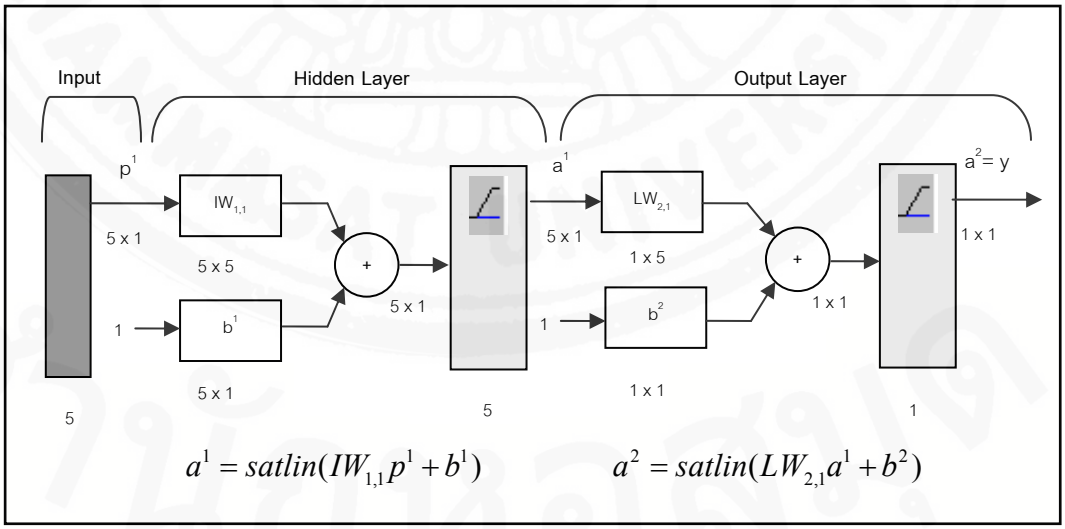
แบบแซทลิน (Saturating Linear Transfer Function : satlin)



ภาพที่ 2.20

แสดงกระบวนการทำงานของโครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับหลายชั้น

มีฟังก์ชันกระตุ้นแบบเชิงเส้นสำหรับงานวิจัยชิ้นนี้



การปรับค่าเวกเตอร์น้ำหนักโดยอัลกอริทึมการแพร่กระจายย้อนกลับนั้น ต้องนิยามค่าผิดพลาดการสอนสำหรับข่ายงาน กำหนดให้แทนค่าผิดพลาดการสอนด้วย $E\left(\vec{w}\right)$ และสามารถคำนวณได้ดังสูตรต่อไปนี้

$$E(\vec{w}) = \frac{1}{2}\sum_{d\in D}\sum_{k\in outputs}(t_{kd} - o_{kd})^2 \qquad (2.13)$$

เมื่อ $outputs$ ค่าเอาต์พุตที่ได้ คือ เซตของบัพเอาต์พุตในโครงข่าย

t_{kd} คือ ค่าเอาต์พุตเป้าหมายของบัพเอาต์พุตที่ k ของตัวอย่างที่ d

o_{kd} คือ ค่าเอาต์พุตที่ได้จากโครงข่ายของบัพเอาต์พุตที่ k ของตัวอย่างที่ d

อัลกอริทึมการแพร่กระจายย้อนกลับจะทำการค้นหาเวกเตอร์น้ำหนักที่ให้ค่าผิดพลาดต่ำที่สุด แต่ในกรณีของข่ายงานป้อนไปข้างหน้าหลายชั้น ค่าต่ำสุดโดยทั่วไปมีมากกว่าหนึ่งจุด ดังนั้น คำตอบที่ได้จากใช้อัลกอริทึมการแพร่กระจายย้อนกลับ จึงเป็น ค่าผิดพลาดต่ำสุดเฉพาะที่

กำหนดตัวอย่างที่ใช้ในการเรียนรู้อยู่ในรูปแบบ $\langle \vec{x}, \vec{t} \rangle$

เมื่อ $\vec{x}$ คือ อินพุตเวกเตอร์ของโครงข่าย

$\vec{t}$ คือ เวกเตอร์เป้าหมายของโครงข่าย

η คือ อัตราการเรียนรู้ (Learning Rate)

$n_{in}, n_{out}, n_{hidden}$ คือ จำนวนของโครงข่ายชั้นอินพุต เอาต์พุต และชั้นซ่อน

x_{ij} คือ อินพุตจากองค์ประกอบ i ไปยังองค์ประกอบ j

w_{ij} คือ น้ำหนักจากองค์ประกอบ i ไปยังองค์ประกอบ j

ขั้นตอนการเรียนรู้ของอัลกอริทึมการแพร่กระจายย้อนกลับ

1. กำหนดค่าเริ่มต้น โดยการแบบสุ่มตัวเลขจำนวนน้อยๆ อาทิเช่น ค่าระหว่าง -0.05 ถึง 0.05
2. ปรับค่าน้ำหนักตามขั้นตอนดังนี้

   สำหรับแต่ละตัวอย่างที่ใช้ในการเรียนรู้ $\langle \vec{x}, \vec{t} \rangle$

   (ก.) อินพุต $\vec{x}$ เข้าสู่โครงข่าย และคำนวณเอาต์พุต o_u สำหรับทุก u

   (ข.) คำนวณค่าความคลาดเคลื่อนของแต่ละค่าเอาต์พุตในโครงข่าย

   $$\delta_k = o_k(1-o_k)(t_k - o_k) \qquad (2.14)$$

   (ค.) คำนวณค่าความคลาดเคลื่อนของแต่ละชั้นซ่อน

$$\delta_h = o_h(1-o_h)\sum_{k\in outputs} w_{kh}\delta_k \qquad (2.15)$$

(ง.) ปรับปรุงค่าน้ำหนักของเส้นเชื่อมในแต่ละโครงข่าย

$$w_{ji} : w_{ji} \leftarrow w_{ji} + \Delta w_{ji} \qquad (2.16)$$

$$\Delta w_{ji} = \eta\delta_j x_{ji} \qquad (2.17)$$

(บุญเสริม กิจศิริกุล, 2546)

กำหนดตัวอย่างที่ใช้ในการเรียนรู้อยู่ในรูปแบบคู่อันดับ $(\vec{x}, \vec{y})$

เมื่อ $\vec{x}$ คือ อินพุตเวกเตอร์ของโครงข่ายประสาทเทียม

$\vec{y}$ คือ เอาต์พุตเป้าหมายของโครงข่ายประสาทเทียม

ฟังก์ชันกระตุ้น (Activation Function) เช่นใช้ฟังก์ชัน ซิกมอยด์ f(y) = $\frac{1}{1+e^{-y}}$

η คือ ค่าอัตราการเรียนรู้ (Learning Rate)

x_{ij} คือ อินพุตจากองค์ประกอบ i ไปยังองค์ประกอบ j

w_{ij} คือ น้ำหนักจากองค์ประกอบ i ไปยังองค์ประกอบ j

ขั้นตอนการเรียนรู้ของโครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับ

1. สร้างโครงข่ายประสาทเทียมตามโครงสร้างที่ออกแบบไว้ และกำหนดจำนวนเซลล์ประสาทของแต่ละชั้น

2. กำหนดค่าน้ำหนักเริ่มต้นแบบสุ่มให้มีค่าน้อยๆ (เช่น ระหว่าง -0.05 ถึง 0.05) ให้แก่เส้นเชื่อมทุกเส้นในโครงข่าย

3. ปรับค่าน้ำหนักด้วยขั้นตอนดังนี้

- นำตัวอย่าง $(\vec{x}, \vec{y})$ มา 1 ตัว จากเซตของตัวอย่างที่ใช้สอนทั้งหมด
- นำตัวอย่าง $\vec{x}$ เข้าสู่ส่วนอินพุตของโครงข่าย
- คำนวณผลรวมขององค์ประกอบชั้นซ่อน

$$net_{pj}^{h} = \sum_{i=1}^{N} w_{ji}^{h} x_{ji} \quad (2.18)$$

(ก) คำนวณเอาต์พุตของผลรวมองค์ประกอบที่ได้ของชั้นซ่อน

$$o_{pj}^{h} = f_{j}^{h}(net_{pj}^{h}) = \frac{1}{1+e^{-net_{pj}^{h}}} \quad (2.19)$$

(ข) คำนวณผลรวมขององค์ประกอบชั้นเอาต์พุต

$$net_{pk}^{o} = \sum_{i=1}^{L} w_{kj}^{o} o_{pj}^{h} \quad (2.20)$$

(ค) คำนวณเอาต์พุตของผลรวมองค์ประกอบชั้นเอาต์พุต

$$o_{pk}^{k} = f_{k}^{o}(net_{pk}^{o}) = \frac{1}{1+e^{-net_{pk}^{o}}} \quad (2.21)$$

(ง) ทำขั้นตอนตั้งแต่ (ข.) ถึง (จ.) สำหรับทุกๆ ชั้นของโครงข่าย

(จ) คำนวณค่าความคลาดเคลื่อน δ_{pk}^{o} สำหรับรูปแบบที่ใช้สอน p กับทุกองค์ประกอบ k ในชั้นเอาต์พุต

$$\delta_{pk}^{o} = o_{k}(1-o_{k})(y_{k}-o_{k}) \quad (2.22)$$

(ฉ) คำนวณค่าความคลาดเคลื่อน δ_{pj}^{h} สำหรับรูปแบบที่ใช้สอน p กับทุกองค์ประกอบ j ในชั้นซ่อน

$$\delta_{pj}^{h} = o_{j}(1-o_{j})\sum_{k=1}^{L} \delta_{pk}^{o} w_{kj} \quad (2.23)$$

(ช) ปรับค่าน้ำหนักของเส้นเชื่อม ให้กับชั้นเอาต์พุต

$$w_{kj}(t+1) = w_{kj}(t) + \Delta w_{kj} \quad (2.24)$$

โดยที่ $\Delta w_{kj} = \eta \delta_{pk}^{o} o_{pj}^{h}$ (2.25)

(ฌ) ปรับค่าน้ำหนักของเส้นเชื่อม ให้กับชั้นซ่อน

$$w_{ji}(t+1) = w_{ji}(t) + \Delta w_{ji} \quad (2.26)$$

โดยที่ $\Delta w_{ji} = \eta \delta^{h}_{pj} x_i$ (2.27)

(ญ) ทำซ้ำขั้นตอน (ข) ถึงขั้นตอน (ฌ) สำหรับทุกคู่ลำดับเวกเตอร์อินพุตในตัวอย่างที่ใช้สอนทั้งเซต โดยจะเรียกว่าเป็นการสอนจำนวน 1 รอบ (Epoch)

(ฏ) ทำซ้ำขั้นตอน (ก) ถึงขั้นตอน (ญ) สำหรับการสอนหลายรอบ เพื่อที่จะทำให้ลดค่าผิดพลาดกำลังสอง ซึ่งการคำนวณหาค่าผิดพลาดกำลังสองใช้สมการดังสมการที่ 2.13 (วรวุฒิ ศรีสุขคำ, 2547)

2.1.2.3 เทคนิคโครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับ

โครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับหลายชั้น มีขั้นตอนวิธีในการเรียนรู้เพื่อปรับค่าน้ำหนักสำหรับโครงข่ายประสาทเทียมแบบหลายชั้น (Multilayer Neural Network) โดยใช้ วิธีการเกรเดียนต์เดสเซนต์ (Gradient Descent) เพื่อลดค่าผิดพลาดกำลังสอง (Square Error) ให้มีค่าน้อยที่สุด กล่าวคือ จะต้องมีค่าผิดพลาดกำลังสองเฉลี่ย (Mean Square Error) ระหว่างเอาต์พุตที่ได้จากโครงข่าย (Network) และเอาต์พุตค่าเป้าหมาย (Target) ต่ำที่สุด (สุกรี สินธุภิญโญ, 2544)

โดยวิธีการสำหรับปรับค่าน้ำหนักของโครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับหลายชั้นนั้นดำเนินการดังสมการต่อไปนี้

$$o = \sigma(\vec{w} \cdot \vec{x}) \quad (2.28)$$

โดย

$$\sigma(y) = \frac{1}{1+e^{-y}} \quad (2.29)$$

เมื่อ o คือ ข้อมูลนำออก หรือเอาต์พุต (Output)

$\vec{x}$ คือ ข้อมูลนำเข้า หรืออินพุต (Input)

$\vec{w}$ คือ ค่าน้ำหนักของอินพุตนั้นๆ (Weight of Input)

σ คือ ฟังก์ชันซิกมอยด์ (Sigmoid Function) ซึ่งให้ค่านำออก หรือเอาต์พุต ระหว่าง 0 ถึง 1

เมื่อกล่าวถึงงานวิจัยที่นำเสนอในวิทยานิพนธ์ฉบับนี้ ได้นำเสนอโครงข่ายประสาทเทียมแบบข่ายงานเพอเซปตรอนหลายชั้น (Mulitlayer Perceptron Neural Network) โดยใช้อัลกอริทึมการแพร่กระจายย้อนกลับ ซึ่งมีรูปแบบการทำงานแบบข่ายงานป้อนไปข้างหน้า (Feed Forward Network) กล่าวคือ ในขั้นตอนการทำงานนั้นจะไม่มีการป้อนผลลัพธ์ที่ได้ในแต่ละบัพ (Node) ย้อนกลับไปยังบัพก่อนหน้านี้ที่ส่งข้อมูลมาให้

ส่วนในขั้นตอนการเรียนรู้ (Learning Process) จะมีการปรับค่าน้ำหนักของแต่ละบัพ เมื่อคำตอบที่ได้มาผิดพลาด โดยการปรับค่าน้ำหนักจะกระทำโดยส่งข้อมูลย้อนกลับไป จึงเรียกอัลกอริทึมการเรียนรู้ในลักษณะนี้ว่า อัลกอริทึมการแพร่กระจายย้อนกลับ (Backpropagation Algorithm) และค่าน้ำหนักที่ได้จะเป็นค่าน้ำหนักที่ทำให้ค่าผิดพลาดกำลังสองเฉลี่ย (Mean Square Error) มีค่าต่ำที่สุด การทำงานของโครงข่ายประสาทเทียมแบบแพร่กระจายย้อนกลับในขั้นตอนการเรียนรู้ นั้น ทำการการสอน (Training) โครงข่ายประสาทเทียม (Neural Network) ให้สามารถเรียนรู้ถึงความสัมพันธ์กันระหว่างอินพุตและเอาต์พุต และการเรียนรู้นั้นจะต้องมีค่าผิดพลาดน้อยที่สุด (สุขวสา พิชิตเดช, 2544)

2.2 งานวิจัยที่เกี่ยวข้อง

ในส่วนของงานวิจัยและบทความที่เกี่ยวข้องกับวิทยานิพนธ์ฉบับนี้ ประกอบไปด้วยงานวิจัยและบทความที่นำเทคนิคการตัดคำมาประยุกต์ใช้กับภาษาไทย งานวิจัยที่มีการประยุกต์ใช้เทคนิคการวิเคราะห์หาความหมายแอบแฝง และงานวิจัยที่มีการประยุกต์ใช้เทคนิคโครงข่ายประสาทเทียม

2.2.1 งานวิจัยและบทความที่นำเทคนิคการตัดคำมาประยุกต์ใช้กับภาษาไทย

งานวิจัยเรื่อง การวิเคราะห์แนวทางการเปรียบเทียบสมรรถนะของโปรแกรมแยกคำภาษาไทย ของ พิสิทธิ์ พรมจันทร์ (2540) เป็นงานวิจัยที่ต้องการจะเปรียบเทียบประสิทธิภาพในแง่มุมต่างๆ ของวิธีการตัดคำที่มีการนำมาประยุกต์ใช้โดยทั่วไปในปัจจุบัน ซึ่งการเปรียบเทียบ

ประสิทธิภาพของการตัดคำนั้นมีประโยชน์ในการพิจารณาเลือกใช้วิธีการตัดคำภาษาไทยให้เหมาะสม และมีประโยชน์ในการพัฒนามาตรฐานการตัดคำภาษาไทยต่อไป

มาตรวัดประสิทธิภาพของขั้นตอนวิธีการตัดคำแบบต่างๆ

- ความสามารถในการที่จะตัดคำได้
- ความถูกต้องของคำหลังจากผ่านการตัดคำแล้ว
- สัดส่วนความถูกต้องของคำที่ตัดออกมาต่อจำนวนคำที่ใช้ในพจนานุกรม
- ความถูกต้องเชิงไวยากรณ์ของประโยคหลังจากผ่านการตัดคำแล้ว
- ความถูกต้องเชิงความหมายของประโยคหลังจากที่ผ่านการตัดคำแล้ว
- ความสามารถที่จะแบ่งวรรคตอนภาษาไทย
- ความสามารถที่จะปรับสระและวรรณยุกต์ที่ติดกันอย่างไม่ถูกต้องให้ถูกต้อง
- มาตรวัดประสิทธิภาพเชิงความเร็ว
- มาตรวัดประสิทธิภาพการใช้ทรัพยากร
- มาตรวัดประสิทธิภาพการแก้ไขความผิดพลาดในการตัดคำ

โปรแกรมการตัดคำภาษาไทยได้มีการพัฒนาและใช้งานกันอย่างกว้างขวางทั้งในด้านการศึกษาวิจัย และการนำไปประยุกต์ใช้จริงในงานด้านต่างๆ ทั้งทางภาครัฐและเอกชน งานวิจัยชิ้นนี้ได้ทำการรวบรวมคุณลักษณะ เทคนิคของโปรแกรมและอัลกอริทึมการตัดคำที่มีการใช้ในขณะที่ทำการศึกษางานวิจัยชิ้นนี้ ดังนี้

- โปรแกรมการตัดคำภาษาไทยแบบย้อนกลับ (Backtracking Algorithm)
- โปรแกรมการตัดคำภาษาไทยแบบการเทียบคำที่ยาวที่สุด (Longest Pattern Matching)
- โปรแกรมการตัดคำภาษาไทยแบบการเทียบคำที่สั้นที่สุด (Shortest Pattern Matching)
- โปรแกรมการตัดคำภาษาไทยแบบที่ใช้ความถี่ของการใช้คำ (Word Usage Frequency)
- โปรแกรมการตัดคำภาษาไทยแบบที่ใช้พจนานุกรมลดความกำกวม (Ambiguity Dictionary)

- โปรแกรมการตัดคำภาษาไทยแบบที่เลือกประโยคที่มีจำนวนคำน้อยที่สุด (Maximal Pattern Matching)

เนื่องจากวิทยานิพนธ์ฉบับนี้ได้นำโปรแกรมการตัดคำภาษาไทยแบบที่เลือกประโยคที่มีจำนวนคำน้อยที่สุด (Unknown Word) หรือที่เรียกกันว่า การตัดคำโดยเลือกแบบเหมือนมากที่สุด (Maximal Matching) มาใช้ในการทำวิจัย จึงขอกล่าวรายละเอียดในส่วนหลักการที่เกี่ยวข้องกับโปรแกรมนี้ว่า สามารถที่จะทำการตัดคำออกมาทุกกรณีที่เป็นไปได้แต่จะเลือกประโยคที่ประกอบไปด้วยจำนวนคำที่น้อยที่สุด โดยงานวิจัยของ พิสิทธิ์ พรมจันทร์ (2540) นี้ได้รับความอนุเคราะห์จากห้องวิจัยลิงค์ (LINKS) ของศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) ให้ใช้โปรแกรมการตัดคำที่ใช้เทคนิคแบบเลือกประโยคที่มีจำนวนคำที่น้อยที่สุด หรือที่เรียกกันว่า การตัดคำโดยเลือกแบบเหมือนมากที่สุด ดังที่กล่าวมาแล้วมาทำการวัดประสิทธิภาพร่วมกันกับโปรแกรมการตัดคำแบบอื่นๆ จากการวิเคราะห์ผลการวัดประสิทธิภาพของโปรแกรมตัดคำภาษาไทยได้ข้อสรุปดังแสดงในตารางที่ 2.7 นี้

ตารางที่ 2.7

สรุปคุณลักษณะข้อเด่น ข้อด้อย และตัวอย่างการนำไปใช้งานที่เหมาะสม

โปรแกรมตัดคำ	ข้อเด่น	ข้อด้อย	งานที่เหมาะสมที่จะนำไปใช้
แบบย้อนรอยกลับ (Back Tracking)	ได้จำนวนคำผิดน้อยที่สุด มีวิธีการกู้คืนจากความผิดพลาด	ได้จำนวนคำทั้งหมดน้อยที่สุด	ตรวจสอบคำสะกด
แบบเทียบคำยาวที่สุด (Longest Matching)	ได้จำนวนคำที่ถูกต้องมากที่สุด		จัดรูปแบบเอกสาร
แบบเทียบคำสั้นที่สุด (Shortest Matching)	ได้จำนวนคำทั้งหมดมากที่สุด	ได้จำนวนคำผิดมากที่สุด ได้จำนวนคำที่ถูกต้องน้อยที่สุด ได้สัดส่วนความถูกต้องต่อจำนวนคำในพจนานุกรมต่ำที่สุด	ไม่เหมาะกับการนำไปประยุกต์ใช้
แบบอาศัยความถี่คำ (Word Frequency)	ได้สัดส่วนความถูกต้องต่อจำนวนคำในพจนานุกรมสูงที่สุด		ต้องไปพัฒนาเพิ่มเติมจึงจะนำไปใช้งานได้
แบบเลือกจำนวนคำน้อยที่สุด (Maximal Matching)			จัดรูปแบบเอกสาร
แบบใช้พจนานุกรมคำกำกวม (Ambiguity Dictionary)	ได้ความผิดพลาดที่เกิดจากความกำกวมน้อยที่สุด		สังเคราะห์เสียงพูดจากประโยคการแปลภาษาด้วยเครื่อง

ที่มา: พิสิทธิ์ พรมจันทร์ (2540)

บทความเรื่อง การตัดคำไทยในระบบแปลภาษา ของ วิรัช ศรเลิศล้ำวาณิช (2536) นำเสนอการนำคอมพิวเตอร์เข้ามาช่วยในการประมวลผลข้อมูลทางภาษา ทางด้านการตัดคำ การแบ่งข้อความที่ต่อเนื่องกันออกเป็นหน่วยคำ (Morpheme) หรือลักษณะของการรู้จำ

(Recognition) คำหนึ่งๆ ในข้อความที่ต่อเนื่องกัน โดยมีจุดประสงค์เพื่อหาขอบเขตของแต่ละหน่วยคำ กล่าวคือ หากสามารถรู้ขอบเขตของแต่ละคำแล้ว การจัดการกับข้อความนั้นๆ ก็จะเป็นไปได้อย่างสะดวกและถูกต้อง ในระบบคอมพิวเตอร์จึงจำเป็นอย่างยิ่งที่จะต้องคำนวณหาขอบเขตของแต่ละคำให้ได้ เพื่อที่จะลดภาระการทำงานของผู้ใช้ หรือเพื่อที่จะให้กระบวนการที่อยู่ในระดับที่ลึกกว่าสามารถทำงานต่อไปได้ อีกทั้งยังอาจจะเป็นตัวช่วยในการกำหนดคำเพื่อทำการวิเคราะห์ในระบบเครื่องแปลภาษาต่อไป

งานวิจัยชิ้นนี้นำเสนออัลกอริทึมสำหรับการตัดคำโดยใช้พจนานุกรม เพื่อหาขอบเขตของหน่วยคำในข้อความที่ต่อเนื่อง ทั้งหมด 3 อัลกอริทึม ดังต่อไปนี้

- กฎทางอักขระวิธี
- วิธีการเทียบคำยาวที่สุด
- วิธีการตัดคำให้ได้จำนวนคำและคำที่ไม่มีในพจนานุกรมน้อยที่สุด หรือที่เรียกกันว่าวิธีการตัดคำโดยเลือกแบบเหมือนมากที่สุด

ในส่วนของรายละเอียดของแต่ละอัลกอริทึม วิทยานิพนธ์ฉบับนี้ได้กล่าวไว้แล้วในส่วนของทฤษฎีที่เกี่ยวข้องหัวข้อการตัดคำ (Word Segmentation)

กล่าวถึงบทสรุป การตัดคำเป็นกระบวนการเริ่มแรกที่จะนำข้อมูลเข้าสู่ระบบการแปลภาษาด้วยเครื่องคอมพิวเตอร์ และต้องการการประมวลผลที่มีความเร็วสูง วิธีดังกล่าวที่ได้นำมาใช้ในงานวิจัยนี้ โดยนำหลักการทั้ง 3 ประการดังกล่าว มาประยุกต์เข้าด้วยกัน จนทำให้ผลลัพธ์ที่ได้มีความถูกต้องสูงถึงร้อยละ 90

2.2.2 งานวิจัยที่มีการประยุกต์ใช้เทคนิคการวิเคราะห์หาความหมายแอบแฝง

งานวิจัยเรื่อง Indexing by Latent Semantic Analysis ของ Deerwester, Dumais และ Harshman (1986) กล่าวถึง ดัชนีการวิเคราะห์หาความหมายแอบแฝง ที่มีการใช้กระบวนการแยกค่าแบบเดี่ยว มาทำการสร้างเมตริกซ์ คำ-เอกสารที่มีความสัมพันธ์กันและทำการสร้าง Semantic Space ซึ่งประกอบไปด้วยค่าแบบเดี่ยวที่ถูกจัดเรียงใน Space สะท้อนให้เห็นถึงรูปแบบความสัมพันธ์ของคำกับเอกสาร และโครงสร้างของเนื้อหาที่มีความเกี่ยวข้องกัน

ในส่วนสำคัญที่นำมาใช้ในวิทยานิพนธ์ฉบับนี้ คือ รายละเอียดทางด้านเทคนิค และการนำเสนอตัวอย่างการใช้เทคนิคการวิเคราะห์หาความหมายแอบแฝง และเทคนิคการวิเคราะห์การแยกค่าแบบเดี่ยว ในการสร้างเมตริกซ์

งานวิจัยเรื่อง How Well Can Passage Meaning be Derived without Using Word Order A Comparison of Latent Semantic Analysis and Humans ของ Landauer, Laham, Rehder และ Schreiner (1987) นำเสนอว่า ความเข้าใจของมนุษย์นั้นเกิดขึ้นจากการเรียงประโยคที่ถูกต้องตรงตามหลักไวยากรณ์ และจำเป็นที่จะต้องมีการเรียงลำดับก่อนหลังอย่างมีความสัมพันธ์กันภายในประโยค หากทำการทดลองสุ่มลำดับของคำเข้ามา จะทำให้ความสามารถในการทำความเข้าใจของมนุษย์และการจับใจความสำคัญของมนุษย์ลดลง แต่ในทางกลับกันกระบวนการทำงานของการค้นคืนข้อมูลในปัจจุบัน นั้นใช้หลักเกณฑ์ทางไวยากรณ์ในการค้นหาข้อมูลน้อยมาก จึงพยายามคิดค้นกระบวนการที่จะทำให้โปรแกรมสามารถเลียนแบบความเข้าใจของมนุษย์ได้ ซึ่งในปัจจุบันการพัฒนาเทคนิคการวิเคราะห์หาความหมายแอบแฝงทำให้สามารถสร้างโปรแกรมที่จำลองความเข้าใจของมนุษย์ได้ตรงตามความต้องการ ซึ่งเทคนิคการวิเคราะห์หาความหมายแอบแฝง สามารถทำการจับคู่คำที่มีความหมายเหมือนกันแต่เขียนไม่เหมือนกันได้ และยังสามารถตัดเอกสารที่มีคำเหมือนกันแต่ความหมายแตกต่างกันออกไปได้อีกด้วย จากการที่ระบบมีการประยุกต์ใช้เทคนิคการวิเคราะห์หาความหมายแอบแฝงโดยทำการลดมิติจะทำให้ประสิทธิภาพในการค้นหาเหนือกว่าระบบที่ใช้การค้นหาคำที่เหมือนกันโดยตรงถึง 3 เท่า

บทสรุปของงานวิจัยนี้กล่าวว่า เทคนิคการวิเคราะห์หาความหมายแอบแฝงสามารถดึงข้อมูลที่อยู่ในบทความออกมาได้อย่างอิสระไม่ขึ้นต่อการเรียงลำดับของคำ สิ่งสำคัญที่ค้นพบคือการประยุกต์ใช้เทคนิคการวิเคราะห์หาความหมายแอบแฝง ในการประเมินผลการเขียนบทความที่ไม่มีการเรียงลำดับของคำได้ผลใกล้เคียงกับการประเมินโดยมนุษย์ และการประยุกต์ใช้เทคนิคการวิเคราะห์หาความหมายแอบแฝง ในการวิเคราะห์หาความหมายแอบแฝงในบทความได้ผลเทียบเท่ากับความสามารถในการวิเคราะห์หาความหมายแอบแฝงโดยมนุษย์ จากผลดังกล่าวเหล่านี้ แสดงให้เห็นได้ว่า เทคนิคการวิเคราะห์หาความหมายแอบแฝง เป็นเทคนิคที่มีความสามารถในการดึงความหมายที่ซ่อนอยู่ในบทความออกมาได้โดยไม่ต้องอาศัยหลักเกณฑ์ทางไวยากรณ์ และสิ่งที่สำคัญก็คือผลลัพธ์ที่ได้จะขึ้นอยู่กับ Semantic Space ที่สร้างจากมิติที่เหมาะสม (Optimal Dimension)

งานวิจัยเรื่อง An Introduction to Latent Semantic Analysis ของ Landauer, Foltz และ Laham (1998) นำเสนอเทคนิคการวิเคราะห์หาความหมายแอบแฝง เป็นทฤษฎี และวิธีการค้นคืนคำตามความหมายของคำที่ต้องการค้นหา โดยประยุกต์การคำนวณทางด้านสถิติเข้ากับกลุ่มของบทความขนาดใหญ่ โดยแนวความคิดพื้นฐานคือการรวบรวมบทความที่มีคำที่ต้องการ

ค้นหาปรากฏอยู่ และบทความที่ไม่มีคำที่ต้องการค้นหาปรากฏอยู่ โดยจัดเป็นกลุ่มของคำขนาดใหญ่เพื่อใช้ในการตัดสินใจภายใต้กฎเกณฑ์ร่วมกันที่จะกำหนดกลุ่มของคำทั้งที่มีความหมายเหมือนกันและกลุ่มของคำที่มีความหมายแตกต่างกัน แสดงให้เห็นว่า เทคนิคการวิเคราะห์หาความหมายแอบแฝง มีความสามารถเพียงพอที่จะสะท้อนให้เห็นถึงองค์ความรู้ของมนุษย์ที่ถูกสร้างขึ้นมาจากหลากหลายแนวทาง ยกตัวอย่างเช่น การที่ระบบทำการให้คะแนนแบบทดสอบความรู้ของมนุษย์ โดยคำนึงถึงคำศัพท์มาตรฐานและเนื้อหาสาระของเรื่องที่ทำการทดสอบ ซึ่งการให้คะแนนด้วยการประยุกต์ใช้เทคนิคการวิเคราะห์หาความหมายแอบแฝง นั้นจะทำการเลียนแบบการค้นหาคำ และการจัดกลุ่มคำอย่างเหมาะสมของมนุษย์ ได้โดย เทคนิคการวิเคราะห์หาความหมายแอบแฝงจะทำการเปรียบเทียบ คำกับคำ และบทความกับคำที่อยู่ในพจนานุกรม และจากนั้นจะรายงานการทดสอบในประเด็นสำคัญ 3 ประเด็น ดังนี้ ประเด็นแรก รายงานความสามารถของเทคนิคการวิเคราะห์หาความหมายแอบแฝงในการประเมินความต่อเนื่องของบทความ ประเด็นที่สอง รายงานความสามารถในการเรียนรู้บทความของผู้ที่ทำการทดสอบ และประเด็นสุดท้าย รายงานคุณภาพและปริมาณองค์ความรู้ในเรียงความที่ทำการทดสอบ

งานวิจัยเรื่อง Learning from text: Matching readers and texts by Latent Semantic Analysis ของ Wolfe, Schreiner, Rehder, Laham, Foltz, Kintsch และ Landauer (1998) นำเสนองานวิจัยที่ต้องการพิสูจน์ระดับความสามารถของผู้อ่านในการดึงองค์ความรู้โดยสรุปออกมาจากบทความว่ามีปัจจัยขึ้นอยู่กับระดับความรู้พื้นฐานของผู้อ่านและระดับความยากง่ายของเนื้อหาของบทความที่ทำการศึกษานั้น โดยมีการนำเทคนิคการวิเคราะห์หาความหมายแอบแฝงมาประยุกต์ใช้เพื่อแสดงให้เห็นถึงข้อสรุปที่ว่าบทความที่ทำให้ผู้อ่านได้รับความรู้มากที่สุดไม่ได้มากจากบทความที่มีระดับความยากที่สุด หรือง่ายที่สุด แต่เกิดจากการเรียนรู้ที่ได้จากการเชื่อมโยงความรู้ที่ได้เรียนรู้ใหม่กับความรู้เก่าที่มีอยู่ก่อนแล้วเพื่อนำมาใช้ในการตีความเพื่อให้เกิดความเข้าใจในการอ่าน และบทความที่มีเนื้อหาไม่ตรงกับความรู้พื้นฐานของผู้อ่านจะไม่มีความเหมาะสมในการที่จะนำมาเป็นบทความในการให้ความรู้ และภายในเนื้อหาของงานวิจัยแสดงให้เห็นว่าเทคนิคการวิเคราะห์หาความหมายแอบแฝงสามารถนำไปประยุกต์ใช้ในการจับคู่ระหว่างความรู้พื้นฐานของผู้อ่านกับระดับความยากง่ายของบทความได้หลากหลายแนวทาง

งานวิจัยเรื่อง Automatic Assessment of the Content of Essays Based on Course Materials ของ Kakkonen และ Sutinen (2003) นำเสนอ การประเมินงานเขียนเรียงความ ด้วย

กระบวนการตรวจให้คะแนนโดยอัตโนมัติบนปัจจัยพื้นฐานที่กำหนด และนำเสนอกระบวนการให้ค่าความพึงพอใจด้วยการประยุกต์ใช้เทคนิคการวิเคราะห์หาความหมายแอบแฝง ซึ่งพัฒนามาจากหลักการของศาสตร์การสืบค้นคืนข้อมูล ซึ่งประสบความสำเร็จในการประยุกต์ใช้กับการประเมินงานเขียนเรียงความในหลากหลายแนวทาง อีกทั้งได้นำเสนอการใช้เทคนิคการวิเคราะห์หาความหมายแอบแฝงมาเพื่อเปรียบเทียบแนวคิดความคล้ายคลึงกันระหว่างงานเขียนเรียงความกับบทความที่ได้มาจากเนื้อหาในบทเรียนซึ่งครอบคลุมหัวข้อรายวิชาที่ต้องการทดสอบ ผลการทดลองที่ได้จากการพิจารณาความสัมพันธ์กัน (Correlation) ระหว่าง คะแนนที่ได้จากระบบกับคะแนนที่ได้จากผู้ประเมินพบว่ามีค่าตั้งแต่ 0.78 ถึง 0.82

งานวิจัยเรื่อง การประเมินคุณภาพการถอดความภาษาไทยด้วยการวิเคราะห์หาความหมายแอบแฝง ของ ศุภวรรณ เปรมกุศลชัย (2548) นำเสนอเทคนิคการวิเคราะห์หาความหมายแอบแฝงมาประยุกต์ใช้กับการประเมินคุณภาพการเขียนถอดความภาษาไทย โดยการทดลองนำบทคัดย่อวิทยานิพนธ์ภาษาไทย จำนวน 26 ฉบับ โดยได้คัดเลือกบทคัดย่อจำนวน 3 ฉบับมาเป็นต้นแบบ ให้นักศึกษาปริญญาโท 53 คนเขียนถอดความ จากนั้นนำงานเขียนถอดความที่ได้ไปประเมินโดยอาจารย์ในระดับอุดมศึกษา 4 ท่าน และนำงานเขียนถอดความไปเข้าระบบที่ทำการประเมินคุณภาพการถอดความภาษาไทยด้วยเทคนิคการวิเคราะห์หาความหมายแอบแฝง จากการทดลองพบว่าค่าสัมประสิทธิ์สหสัมพันธ์ (Spearman's rank correlation coefficient) ของการประเมินที่ได้จากระบบกับการประเมินที่ได้จากผู้ประเมิน อยู่ระหว่าง $r_s = 0.352$ ($p = 0.0749$) ถึง $r_s = 0.812$ ($p<0.0001$) ถ้าผลการประเมินของผู้ประเมินแต่ละท่านเป็นไปในทิศทางเดียวกัน ระบบการประเมินคุณภาพการถอดความภาษาไทยก็จะมีความสอดคล้องกันกับผลที่ได้จากผู้ประเมิน ดังนั้นงานวิจัยนี้แสดงให้เห็นว่าเทคนิคการวิเคราะห์หาความหมายแอบแฝงมีศักยภาพที่จะนำมาพัฒนาทำระบบประเมินการเขียนถอดความที่เป็นภาษาไทยได้

งานวิจัยเรื่อง การย่อความเอกสารภาษาไทยโดยการวิเคราะห์หาความหมายที่แอบแฝง ของ อัษฎางค์ แตงไทย (2548) งานวิจัยนี้ใช้การวิเคราะห์หาความหมายที่แอบแฝง (Latent Semantic Analysis) และอัลกอริทึมการแยกค่าแบบเดี่ยว (Singular Value Decomposition) สำหรับการย่อความเอกสารแบบทั่วไป (Generic Summarization) จะทำการเลือกย่อหน้าที่แสดงใจความสำคัญของเอกสารออกมาจากข้อความ (Paragraph Extraction) แต่สำหรับการย่อความเอกสารตามความต้องการของผู้ใช้แบบคำถาม - คำตอบ (Question - Answer Summarization)

โดยไม่ใช้ความรู้ทางลึก (Deep Understand) จะทำการเลือกข้อความสั้น (Short Passage) และข้อความยาว (Long Passage) งานวิจัยนี้มีจุดประสงค์เพื่อประเมินความสามารถในการย่อความเอกสารของระบบที่พัฒนาขึ้นมา ทำการประเมินได้โดยนำผลการประเมินของระบบมาเปรียบเทียบกันกับผลการประเมินที่ได้จากการตรวจเอกสารของผู้เชี่ยวชาญ จนได้ข้อสรุปว่าการวิเคราะห์หาความหมายที่แอบแฝง โดยใช้อัลกอริทึมการแยกค่าแบบเดี่ยว นั้นสามารถทำให้การเลือกใจความสำคัญทั้งการย่อความแบบทั่วไป (Generic Summarization) มีการจัดลำดับและช่วยเลือกย่อหน้าที่เป็นย่อหน้าที่สำคัญ อีกทั้งยังช่วยลดความซ้ำซ้อนของข้อมูล และในส่วนการย่อความแบบตามความต้องการของผู้ใช้แบบคำถาม - คำตอบ (Question - Answer Summarization) ก็ช่วยให้ทำการเลือกข้อความแบบสั้น (Short Passage) และข้อความแบบยาว (Long Passage) ที่มีความสัมพันธ์กับข้อคำถามได้เป็นอย่างดี สำหรับประสิทธิภาพโดยรวมของการย่อเอกสารแบบทั่วไป (Generic Summarization) และการย่อเอกสารแบบคำถาม-คำตอบ (Question-Answer Summarization) ใช้มาตรการเอฟหนึ่ง และ MRAR มีค่าเท่ากับ 71.11% และ 74.6% ตามลำดับ สำหรับการทดสอบพารามิเตอร์ที่ส่งผลต่อการวิเคราะห์หาความหมายที่แอบแฝง ผลการทดสอบพบว่า การตัดคำแบบ tri-gram, คำหยุดที่รวมตัวเลข 0-9 และช่องว่าง (Space), การลบคำที่ปรากฏน้อยกว่า 1 ครั้งในแต่ละย่อหน้า, การปรับเปลี่ยนความถี่ของเมตริกซ์โดยใช้น้ำหนักของคำ แบบ log-entropy และ การลดขนาดเมตริกซ์โดยค่า Relative Change 20% ให้ผลลัพธ์ได้ดีที่สุด

2.2.3 งานวิจัยที่มีการประยุกต์ใช้เทคนิคโครงข่ายประสาทเทียม

งานวิจัยเรื่อง การโปรแกรมตรรกะเชิงอุปนัยและแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กในการรู้จำตัวพิมพ์อักษรไทย ของ สุกรี สินธุภิญโญ (2541) นำเสนอวิธีการรู้จำโดยโปรแกรมตรรกะเชิงอุปนัย (Inductive Logic Programming: ILP) หรือ ไอแอลพี ร่วมกันกับวิธีการประมาณเพื่อเลือกกฎโดยแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก(Backpropagation Neural Network:BNN) หรือ บีเอ็นเอ็น เพื่อเลือกกฎที่ใกล้เคียงกันกับตัวอย่างในกรณีที่ตัวอย่างนั้นมีสัญญาณรบกวน จากการทดลองบีเอ็นเอ็นแบบที่ 1 ใช้จำนวนสัญพจน์ (Literal) ที่ไม่ตรงกับตัวอย่าง และจำนวณสัญพจน์ที่ตรงกับตัวอย่าง เป็นอินพุตเวกเตอร์ ทำให้ในกระบวนการเรียนรู้ อัตราการรู้จำของบีเอ็นเอ็นแบบที่ 1 นี้มีค่า 92.55% ซึ่งสูงกว่าอัตราการรู้จำของไอแอลพีที่มีอัตราการเรียนรู้ 84.97% แต่วิธีการบีเอ็นเอ็นแบบที่ 1 นี้มีข้อเสียคือ ให้ความสำคัญกับทุกสัญพจน์ในแต่ละกฎเทียบเท่ากัน จึงไม่สามารถ

กำหนดน้ำหนักให้กับสัญพจน์ที่มีความสำคัญมากกว่าได้ ดังนั้นจึงมีการทดลองบีเอ็นเอ็นแบบที่ 2 ซึ่งใช้ค่าความจริงของสัญพจน์ในกฎแต่ละข้อ เพื่อแทนจำนวนสัญพจน์ที่ไม่ตรงกับตัวอย่าง และจำนวนสัญพจน์ที่ตรงกับตัวอย่าง เป็นอิตพุตเวกเตอร์ ทำให้ได้อัตราการรู้จำสูงขึ้นเป็น 94.26% ซึ่งสูงกว่าวิธีการแบบอื่นๆ ที่ทำการทดสอบในงานวิจัยเรื่องนี้

งานวิจัยเรื่อง การประมาณกฎจากวิธีการโปรแกรมตรรกะเชิงอุปนัยด้วยวิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กในการรู้จำตัวพิมพ์อักษรไทย ของ สุกรี สินธุภิญโญ (2544) นำเสนอแนวทางการแก้ปัญหาของวิธีการโปรแกรมตรรกะเชิงอุปนัย หรือ ไอแอลพี เนื่องจากว่ากฎที่ได้จากกระบวนการไอแอลพีไม่สามารถจำแนกตัวอย่างใหม่ หรือตัวอย่างที่มีสัญญาณรบกวนได้ ทั้งนี้เป็นเพราะว่า ไอแอลพีจะทำการเลือกกฎที่ตรงกับตัวอย่างแล้วทำการจำแนกตามกฎนั้น หากไม่มีกฎที่ตรงกับตัวอย่าง ระบบไอแอลพีจะไม่สามารถทำการจำแนกตัวอย่างนั้นได้ และเพื่อแก้ปัญหาดังกล่าวงานวิจัยชิ้นนี้จึง สร้างขั้นตอนวิธีดึงลักษณะสำคัญ (Feature Extraction Algorithm) เพื่อใช้ร่วมกันกับแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก เพื่อใช้ในการประมาณกฎที่ใกล้เคียงกันกับตัวอย่าง จากการทดลองพบว่า เปอร์เซ็นต์ความถูกต้องเพิ่มมากขึ้นจากการใช้กฎไอแอลพีเพียงอย่างเดียวในการจำแนกตัวอย่าง โดยเฉพาะปัญหาที่มีลักษณะเป็นหลายกลุ่ม (Multi-Class Problem) เปอร์เซ็นต์ความถูกต้องสูงกว่าปัญหาลักษณะอื่นที่ได้เปรียบเทียบไว้ในงานวิจัยชิ้นนี้ อีกทั้งยังทำการทดลองความทนทานต่อสัญญาณรบกวนในกรณีที่ชุดข้อมูลแบ่งเป็นสองกลุ่ม ผลการทดลองก็พบว่าเปอร์เซ็นต์ความถูกต้องของวิธีการที่สร้างขึ้นในงานวิจัยชิ้นนี้ ลดลงช้ากว่ากฎที่จากระบบไอแอลพี

งานวิจัยเรื่อง การรู้จำตัวอักษรพิมพ์ภาษาไทยโดยการใช้กลุ่มก้อนของนิวรอลเน็ตเวิร์ก ของ สุขวสา พิชิตเดช (2544) นำเสนอการใช้กลุ่มก้อนของนิวรอลเน็ตเวิร์กอย่างเหมาะสมในการรู้จำตัวพิมพ์อักษรภาษาไทย ซึ่งวิธีการที่นำเสนอเป็นวิธีการรวมผลลัพธ์แบบถ่วงน้ำหนักที่เหมาะสมสำหรับตัวแยกแยะที่ให้ความถูกต้องสูง จากการทดลองได้อัตราการรู้จำที่มีความผิดพลาดต่ำที่สุด คือ 1.53% และ 1.29% สำหรับข้อมูลทดสอบชุดที่ 1 และ 2 ตามลำดับ

งานวิจัยเรื่อง การจำแนกจุดเชื่อมในตัวอักษรพิมพ์ไทยด้วยวิธีการโปรแกรมตรรกะแบบอุปนัยและแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก ของ ลือพล พิพานเมฆาภรณ์ (2546) นำเสนอวิธีการตัดแยกตัวอักษรภาษาไทยที่ติดกันในแนวนอน โดยได้ทำการค้นหาและดึงลักษณะที่สำคัญจากรูปแบบของจุดเชื่อมระหว่างตัวอักษร ทั้งนี้ได้อาศัยวิธีการเรียนรู้จากโปรแกรมตรรกะเชิงอุปนัย เพื่อ

สร้างกฎในการจำแนกจุดเชื่อมระหว่างตัวอักษรดังกล่าว จากนั้นจึงนำกฎที่ได้ไปทำการหาคุณสมบัติสำคัญโดยใช้แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก เพื่อทำให้กฎที่ได้มีความยืดหยุ่นในการใช้งานเพิ่มมากขึ้น และจากผลการทดลองพบว่า กฎที่ได้สามารถทำการจำแนกจุดเชื่อมระหว่างตัวอักษรได้ถูกต้อง 94.94% ซึ่งผลลัพธ์ที่ได้สามารถนำไปเข้าสู่กระบวนการตัดแยกตัวอักษร และกระบวนการรู้จำตัวอักษรต่อไป

งานวิจัยเรื่องการทำนายลักษณะการเคลื่อนไหวของข้อต่อขามนุษย์โดยใช้นิวรอลเน็ตเวิร์ก ของ วรวุฒิ ศรีสุขคำ (2547) นำเสนอ การทำนายลักษณะการเคลื่อนไหวของข้อต่อขาสำหรับส่วนข้อเข่าด้านซ้ายและส่วนข้อเท้าด้านซ้ายในการก้าวเดินไปข้างหน้า โดยการศึกษาวิจัยแบ่งออกเป็น 2 ส่วน ประกอบไปด้วย ส่วนที่ 1 เปรียบเทียบและคัดเลือกโครงสร้างนิวรอลเน็ตเวิร์ก 2 แบบ คือ นิวรอลเน็ตเวิร์กแบบมัลติเลเยอร์เพอเซปตรอน และนิวรอลเน็ตเวิร์กแบบหน่วงเวลา ที่ให้ผลการทำนายองศาการเคลื่อนไหวของข้อต่อขาผิดพลาดเฉลี่ยน้อยที่สุด และส่วนที่ 2 เป็นการปรับค่าตัวแปรของนิวรอลเน็ตเวิร์ก ผลจากการศึกษาแสดงให้เห็นว่า นิวรอลเน็ตเวิร์กแบบหน่วงเวลาที่มีฟังก์ชันกระตุ้นแบบเชิงเส้น ให้ผลการทำนายถูกต้องมากกว่า เน็ตเวิร์กแบบมัลติเลเยอร์เพอเซปตรอน และภายหลังนำนิวรอลเน็ตเวิร์กแบบหน่วงเวลาที่ถูกคัดเลือกมาแล้วมาทำการปรับค่าตัวแปรให้สามารถทำนายลักษณะการเคลื่อนไหวข้อเข่าด้านซ้ายและข้อเท้าด้านซ้ายให้มีความถูกต้องมากขึ้น โดยมีค่าองศาผิดพลาดเฉลี่ยอยู่ในช่วง 6.156 $\pm$0.875 องศา และในช่วง 6.152 $\pm$ 0.780 องศา ที่ระดับความเชื่อมั่นร้อยละ 95 โดยเนื้อหาของงานวิจัยได้ให้รายละเอียดเกี่ยวกับโครงสร้างและการออกแบบของโครงข่ายประสาทเทียม และการปรับค่าตัวแปรที่ส่งผลต่อประสิทธิภาพการทำงานของโครงข่ายประสาทเทียม เทคนิคการหยุดแบบเร็วกว่ากำหนด (Early Stopping) งานวิจัยชิ้นนี้ก็ได้นำการออกแบบโครงสร้างของโครงข่ายประสาทเทียมที่ใช้ฟังก์ชันกระตุ้นแบบเชิงเส้น โดยออกแบบให้มีบัพเอาต์พุตเป็นฟังก์ชันเดียว คือมีบัพเอาต์พุตจำนวนบัพเดียวในการให้ผลลัพธ์