บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 การสรุปใจความสำคัญ

การสรุปใจความสำคัญเป็นการสรุปประเด็นหรือเนื้อหาที่สำคัญของแหล่งข้อมูลเพื่อให้ผู้อ่านได้รับทราบประเด็นหรือเนื้อหาในส่วนที่เป็นใจความสำคัญโดยไม่จำเป็นต้องอ่านเนื้อหาทั้งหมดจากแหล่งข้อมูลเหล่านั้น ทำให้ผู้อ่านสามารถรับทราบข้อมูลข่าวสารต่าง ๆ ได้อย่างครบถ้วนและรวดเร็ว

แนวทางการสรุปใจความสำคัญแบ่งออกเป็น 3 แนวทาง คือ

(1) Frequency-based เป็นการสรุปใจความสำคัญโดยพิจารณาจากความถี่และตำแหน่งของคำต่าง ๆ ในเอกสารเป็นหลัก เช่น ความถี่ของคำสำคัญ ระยะห่างระหว่างคำสำคัญในประโยค เป็นต้น

(2) Knowledge-based เป็นการสรุปใจความสำคัญโดยใช้เทคนิค Classification ซึ่งโครงสร้างหลักของเอกสารที่ได้จะขึ้นอยู่กับการตีความจากฐานความรู้ที่มีอยู่

(3) Discourse-based เป็นการสรุปใจความสำคัญโดยพิจารณาจากความเข้ากันได้และความสอดคล้องกันของประโยคต่าง ๆ โดยยึดตามโครงสร้างประโยคของภาษานั้น ๆ

วิธีการสรุปใจความสำคัญแบ่งออกเป็น 2 วิธี คือ

(1) คัดเลือกจากเอกสารต้นฉบับ (Extraction) ซึ่งแบ่งออกเป็น 3 ขั้นตอน คือ

(1.1) การหาคำสำคัญของเอกสาร

(1.2) การหาขอบเขตของประโยค

(1.3) การคัดเลือกประโยคมาจัดทำเป็นข้อสรุป

(2) จัดทำเป็นบทคัดย่อ (Abstraction) ซึ่งแบ่งออกเป็น 4 ขั้นตอน คือ

(2.1) เอกสารต้นฉบับจะถูกนำไปแปลเพื่อให้ระบบทราบความหมายที่จะเป็นตัวแทนของเอกสาร

(2.2) ตัวแทนของเอกสารจะถูกนำไปใช้ในการแยกข้อมูลที่เป็นใจความสำคัญ ซึ่งรวมถึงหัวเรื่องของเอกสารด้วย

(2.3) ข้อมูลเหล่านั้นจะถูกกลั่นกรองและกำจัดข้อมูลที่ซ้ำซ้อนออกไป

(2.4) ข้อมูลที่ได้จะถูกนำเสนอต่อผู้อ่านในรูปของบทคัดย่อ[2]

2.1.1 งานวิจัยที่เกี่ยวกับการสรุปใจความสำคัญในแนวทาง Frequency-based

2.1.1.1 งานวิจัยของ Yu, L., Liu, M., Ren, F., and Kuroiwa, S.

งานวิจัยของ Yu, L., Liu, M., Ren, F., and Kuroiwa, S.[12] นำเสนอวิธีการสรุปเอกสารภาษาจีนโดยพิจารณาจากความถี่ของคำที่ปรากฏอยู่ในเอกสาร ความยาวของประโยค และตำแหน่งที่ประโยคปรากฏอยู่ในเอกสาร ตามขั้นตอนดังต่อไปนี้

(1) คำนวณหาค่าความสำคัญของคำ w ในประโยค s จากสมการ 2.1

$$W = n \times [\log (M / m)] \qquad \text{สมการ 2.1}$$

โดยที่ n คือ ความถี่ของ w ที่ปรากฏอยู่ใน s, M คือ จำนวนประโยคทั้งหมดในเอกสาร และ m คือ จำนวนประโยคที่มี w ปรากฏอยู่

(2) คำนวณหาค่าความสำคัญของประโยค s จากสมการ 2.2

$$l = \alpha \frac{\sum_{w \in s} W}{\max_{s \in C} \left\{ \sum_{w \in s} W \right\}} + \beta \frac{Length_s}{\max_{s \in C} \left\{ Length_s \right\}} + \gamma\, position_s \qquad \text{สมการ 2.2}$$

โดยที่ C คือ เซตของประโยคทั้งหมดในเอกสาร, w คือ คำที่ปรากฏอยู่ในเอกสาร, $Length_s$ คือ ความยาวของ s, $position_s$ คือ ตำแหน่งที่ s ปรากฏอยู่ในเอกสาร, $\alpha = \beta = 0.3$ และ $\gamma = 0.4$

(3) เลือกประโยคที่เป็นใจความสำคัญโดยพิจารณาจากค่าความสำคัญของประโยคที่คำนวณได้จากสมการ 2.2 เรียงลำดับจากมากไปหาน้อยตามจำนวนที่ต้องการ

จากการทดลองที่ใช้เอกสารข่าวภาษาจีนจำนวน 30 ฉบับ เป็นชุดข้อมูลในการทดลอง โดยให้นักศึกษาจำนวน 3 คน เป็นผู้จัดทำข้อสรุปของเอกสารแต่ละฉบับที่มีอัตราส่วนการบีบอัด 10% และ 20% แล้วนำไปเปรียบเทียบกับข้อสรุปที่จัดทำขึ้นโดยวิธีการที่นำเสนอเพื่อหาความเที่ยงของข้อสรุป พบว่าข้อสรุปที่มีอัตราส่วนการบีบอัด 10% มีความเที่ยง 77% และข้อสรุปที่มีอัตราส่วนการบีบอัด 20% มีความเที่ยง 74%

2.1.1.2 งานวิจัยของ Suwanno, N., Suzuki, Y., and Yamazaki, H.

งานวิจัยของ Suwanno, N., Suzuki, Y., and Yamazaki, H.[9] นำเสนอวิธีการสรุปเอกสารภาษาไทยโดยพิจารณาจากคำประสมที่ปรากฏอยู่ในเอกสาร และชื่อเรื่องของเอกสาร ตามขั้นตอนดังต่อไปนี้

(1) คำนวณหาค่าน้ำหนักของคำ k ในเอกสาร i จากสมการ 2.3

$$w_{i,k} = tf_{i,k} \times [\log (D / n_k)] \qquad \text{สมการ 2.3}$$

โดยที่ $tf_{i,k}$ คือ ความถี่ของ k ที่ปรากฏอยู่ใน i, D คือ จำนวนเอกสารทั้งหมด และ n_k คือ จำนวนเอกสารที่มี k ปรากฏอยู่

(2) คำนวณหาค่าน้ำหนักของย่อหน้า p ในเอกสาร i จากสมการ 2.4

$$P_{tf\text{-}idf}(p) = \frac{\sum_{\{k\} \in P_p} w_{i,k}}{\sum n_p} \qquad \text{สมการ 2.4}$$

โดยที่ $\{k\} \in P_p$ คือ เซตของคำที่ปรากฏอยู่ใน p และ n_p คือ จำนวนคำทั้งหมดใน p

(3) คำนวณหาค่าน้ำหนักของชื่อเรื่อง H_i ในย่อหน้า p จากสมการ 2.5

$$P_{hl}(p) = \frac{\sum_{\{k\} \in H_i \cap P_p} w_{i,k}}{\sum_{\{k\} \in H_i} w_{i,k}} \qquad \text{สมการ 2.5}$$

โดยที่ $\{k\} \in H_i \cap P_p$ คือ เซตของคำที่ปรากฏอยู่ทั้งใน H_i และ Pp และ $\{k\} \in H_i$ คือ เซตของคำที่ปรากฏอยู่ใน H_i

(4) คำนวณหาค่าน้ำหนักรวมของย่อหน้า p จากสมการ 2.6

$$P_{score}(p) = P_{tf\text{-}idf}(p) + P_{hl}(p) \qquad \text{สมการ 2.6}$$

(5) เลือกย่อหน้าที่เป็นใจความสำคัญโดยพิจารณาจากค่าน้ำหนักรวมของย่อหน้าที่คำนวณได้จากสมการ 2.6 เรียงลำดับจากมากไปหาน้อยตามจำนวนที่ต้องการ

จากการทดลองที่ใช้เอกสารข่าวภาษาไทยจำนวน 50 ฉบับ เป็นชุดข้อมูลในการทดลอง โดยให้นักศึกษาจำนวน 4 คน เป็นผู้จัดทำข้อสรุปของเอกสารแต่ละฉบับที่มีอัตราส่วนการบีบอัด 15%, 30% และ 45% แล้วนำไปเปรียบเทียบกับข้อสรุปที่จัดทำขึ้นโดยวิธีการที่นำเสนอเพื่อหา F-score ของข้อสรุป พบว่าข้อสรุปที่มีอัตราส่วนการบีบอัด 15%, 30% และ 45% มี F-score ประมาณ 0.26, 0.43 และ 0.44 ตามลำดับ

2.1.2 งานวิจัยที่เกี่ยวกับการสรุปใจความสำคัญในแนวทาง Knowledge-based

2.1.2.1 งานวิจัยของ Thangthai, A., and Jaruskulchai, C.

งานวิจัยของ Thangthai, A., and Jaruskulchai, C.[10] นำเสนอวิธีการสรุปเอกสารภาษาไทยด้วยวิธี Singular Value Decomposition ตามขั้นตอนดังต่อไปนี้

(1) ทำการตัดคำหยุด เช่น ที่ นั้น ใน ตาม และ เพื่อ ไว้ เป็นต้น ออกจากเอกสาร

(2) ทำการตัดคำแบบ N-Gram ด้วยการเพิ่มช่องว่างให้กับข้อความที่ต้องการตัดคำทั้งด้านหน้าและด้านหลัง โดยมีข้อกำหนดว่าคำที่ประกอบไปด้วยสระและวรรณยุกต์จะมีค่าเท่ากับ 1 Gram เช่นเดียวกับคำที่มีตัวอักษรเพียงตัวเดียว ดังตัวอย่างต่อไปนี้

bi-grams : _ส, สวั, วัส, สดี, ดี_

tri-grams : _สวั, สวัส, วัสดี, สดี_, ดี_ _

quad-grams : _สวัส, สวัสดี, วัสดี_, สดี_ _, ดี _ _ _

(3) ทำการสร้าง Matrix A ซึ่งมีขนาด [M × N] โดยที่ M คือ จำนวนคำทั้งหมดจากการตัดคำแบบ N-Gram, N คือ จำนวนย่อหน้าทั้งหมดในเอกสาร และค่าของสมาชิกใน Matrix คือ ความถี่ของคำที่ปรากฏอยู่ในย่อหน้านั้น ๆ

(4) คำนวณหา Matrix U, S และ V^T จากสมการ 2.7

$$A = U\sum V^T$$

$$= (U_1|U_2|\ldots U_n)\begin{pmatrix} S_1 & & \\ & \ddots & \\ & & S_n \end{pmatrix}\begin{pmatrix} V_1 \\ \vdots \\ V_n \end{pmatrix} \qquad \text{สมการ 2.7}$$

โดยที่ Matrix U มีขนาด [M × M], Matrix S คือ Diagonal Matrix ซึ่งมีขนาด [M × N] และ Matrix V^T มีขนาด [N × N]

(5) เลือกย่อหน้าที่เป็นใจความสำคัญโดยพิจารณาจากค่าสมาชิกของ Matrix V^T เรียงลำดับจากมากไปหาน้อยตามจำนวนที่ต้องการ

จากการทดลองที่ใช้เอกสารข่าวภาษาไทยจำนวน 30 ฉบับ เป็นชุดข้อมูลในการทดลอง โดยให้นักศึกษาจำนวน 1 คน เป็นผู้จัดทำข้อสรุปของเอกสารแต่ละฉบับที่มีอัตราส่วนการบีบอัด 20% และ 30% แล้วนำไปเปรียบเทียบกับข้อสรุปที่จัดทำขึ้นโดยวิธีการที่นำเสนอเพื่อหาความเที่ยงของข้อสรุป พบว่าข้อสรุปที่มีอัตราส่วนการบีบอัด 20% มีความเที่ยงประมาณ 49% และข้อสรุปที่มีอัตราส่วนการบีบอัด 30% มีความเที่ยงประมาณ 46%

2.1.2.2 งานวิจัยของ Witte, R., and Bergler, S.

งานวิจัยของ Witte, R., and Bergler, S.[11] นำเสนอวิธีการสรุปเอกสารภาษาอังกฤษโดยใช้ Fuzzy Clustering Algorithm ตามขั้นตอนดังต่อไปนี้

Require: Initialized cluster graph $G = (V, E)$
1. **for all** np $\in V$ **do**
2. createCluster(np); {create initial clusters}
3. **for all** $e \in E$ **do**
4. **if** $e_\gamma \geq \theta$ then {join clusters?}
5. $c_1 \leftarrow$ getCluster(e_{start});
6. $c_2 \leftarrow$ getCluster(e_{end});
7. mergeClusters(c_1, c_2);

จากการทดลองที่ใช้เอกสารภาษาอังกฤษจำนวน 50 ฉบับ เป็นชุดข้อมูลในการทดลอง และคำนวณหา ROUGE-1 score เพื่อหาประสิทธิภาพของข้อสรุป พบว่าข้อสรุปที่ได้มี ROUGE-1 score อยู่ระหว่าง 0.15 – 0.40

2.1.3 งานวิจัยที่เกี่ยวกับการสรุปใจความสำคัญในแนวทาง Discourse-based

2.1.3.1 งานวิจัยของ Jaruskulchai, C., and Kruengkrai, C.

งานวิจัยของ Jaruskulchai, C., and Kruengkrai, C.[5] นำเสนอวิธีการสรุปเอกสารภาษาไทยโดยพิจารณาจากคำสำคัญของเอกสารและความสัมพันธ์ของย่อหน้าต่าง ๆ ในเอกสาร ตามขั้นตอนดังต่อไปนี้

(1) ทำการตัดคำหยุด เช่น ที่ นั้น ใน ตาม และ เพื่อ ไว้ เป็นต้น ออกจากเอกสาร

(2) นำคำต่าง ๆ ที่เหลืออยู่ซึ่งปรากฏในเอกสารมาจัดทำเป็นชุดคำศัพท์ของเอกสาร

(3) คำนวณหาค่าความคล้ายคลึงกันระหว่างย่อหน้าต่าง ๆ จากสมการ 2.8

$$sim(s_i, s_j) = \frac{\sum_{k=1}^{t} w(s_{i,k})\, w(s_{j,k})}{\sqrt{\sum_{k=1}^{t} [w(s_{i,k})]^2 \sum_{k=1}^{t} [w(s_{j,k})]^2}} \qquad \text{สมการ 2.8}$$

โดยที่ s_i คือ ย่อหน้า i ซึ่งประกอบไปด้วยคำต่าง ๆ จำนวน k คำ $\{w(s_{i,1}), \ldots, w(s_{i,k})\}$ และ s_j คือ ย่อหน้า j ซึ่งประกอบไปด้วยคำต่าง ๆ จำนวน k คำ $\{w(s_{j,1}), \ldots, w(s_{j,k})\}$

(4) คำนวณหาค่าน้ำหนักของคำศัพท์แต่ละคำด้วยเทคนิค TLTF (Term Length Term Frequency) เพื่อหาคำสำคัญของเอกสาร

(5) เลือกย่อหน้าที่เป็นใจความสำคัญโดยพิจารณาจากค่าน้ำหนักของคำสำคัญของเอกสารและค่าความคล้ายคลึงกันระหว่างย่อหน้าต่าง ๆ เรียงลำดับจากมากไปหาน้อยตามจำนวนที่ต้องการ

จากการทดลองที่ใช้เอกสารข่าวภาษาไทยจำนวน 30 ฉบับ เป็นชุดข้อมูลในการทดลอง โดยให้นักศึกษาจำนวน 1 คน เป็นผู้จัดทำข้อสรุปของเอกสารแต่ละฉบับที่มีอัตราส่วนการบีบอัด 20% และ 30% แล้วนำไปเปรียบเทียบกับข้อสรุปที่จัดทำขึ้นโดยวิธีการที่นำเสนอเพื่อหาความเที่ยงของข้อสรุป พบว่าข้อสรุปที่มีอัตราส่วนการบีบอัด 20% มีความเที่ยงประมาณ 55% และข้อสรุปที่มีอัตราส่วนการบีบอัด 30% มีความเที่ยงประมาณ 51%

2.2 คลังข้อมูลออร์คิด

คลังข้อมูลออร์คิด คือ คลังข้อมูลที่สร้างขึ้นจากรายงานการประชุมทางวิชาการของศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติซึ่งประกอบไปด้วยจำนวนประโยคทั้งหมดประมาณ 25,000 ประโยค โดยมีนักภาษาศาสตร์เป็นผู้ที่ทำการแบ่งประโยคแต่ละประโยคออกเป็นคำต่าง ๆ และมีการระบุชนิดของคำให้กับคำต่าง ๆ เหล่านั้นด้วย ดังตัวอย่างต่อไปนี้[1]

เพื่อให้งานวิจัยและพัฒนาของท่านมีคุณค่า และคุณประโยชน์ต่อประเทศชาติยิ่งขึ้นไป//
เพื่อให้/JSBR
งานวิจัยและพัฒนา/NCMN
ของ/RPRE
ท่าน/PPRS
มี/VSTA
คุณค่า/NCMN
<space>/PUNC
และ/JCRG
คุณประโยชน์/NCMN
ต่อ/RPRE
ประเทศชาติ/NCMN
ยิ่งขึ้น/ADVN
ไป/XVAE

ในคลังข้อมูลออร์คิดจะมีชนิดของคำที่มีการระบุให้กับคำต่าง ๆ ทั้งหมด 47 ชนิด แสดงดังตาราง 2.1

ตาราง 2.1 ชนิดของคำภาษาไทยในคลังข้อมูลออร์คิด[1][8]

ลำดับที่	ชนิดของคำ	รายละเอียด	ตัวอย่างคำ
1	NPRP	Proper noun	วินโดวส์ 95, โคโรน่า, โค้ก, พระอาทิตย์
2	NCNM	Cardinal number	หนึ่ง, สอง, สาม, 1, 2, 3
3	NONM	Ordinal number	ที่หนึ่ง, ที่สอง, ที่สาม, ที่1, ที่2, ที่3
4	NLBL	Label noun	1, 2, 3, 4, ก, ข, ค, ง
5	NCMN	Common noun	หนังสือ, อาหาร, อาคาร, คน
6	NTTL	Title noun	ดร., พลเอก
7	PPRS	Personal pronoun	คุณ, เขา, ฉัน
8	PDMN	Demonstrative pronoun	นี่, นั่น, ที่นั่น, ที่นี่
9	PNTR	Interrogative pronoun	ใคร, อะไร, อย่างไร
10	PREL	Relative pronoun	ที่, ซึ่ง, อัน, ผู้
11	VACT	Active verb	ทำงาน, ร้องเพลง, กิน
12	VSTA	Stative verb	เห็น, รู้, คือ
13	VATT	Attributive verb	อ้วน, ดี, สวย
14	XVBM	Pre-verb auxiliary, before negator “ไม่”	เกิด, เกือบ, กำลัง
15	XVAM	Pre-verb auxiliary, after negator “ไม่”	ค่อย, น่า, ได้
16	XVMM	Pre-verb, before or after negator “ไม่”	ควร, เคย, ต้อง
17	XVBB	Pre-verb auxiliary, in imperative mood	กรุณา, จง, เชิญ, อย่า, ห้าม
18	XVAE	Post-verb auxiliary	ไป, มา, ขึ้น
19	DDAN	Definite determiner, after noun without classifier in between	นี่, นั่น, โน่น, ทั้งหมด

ตาราง 2.1 (ต่อ)

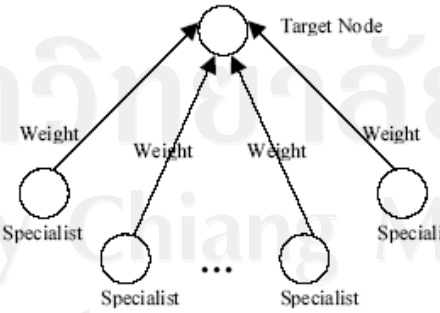
ลำดับที่	ชนิดของคำ	รายละเอียด	ตัวอย่างคำ
20	DDAC	Definite determiner, allowing classifier in between	นี้, นั้น, โน้น, นู้น
21	DDBQ	Definite determiner, between noun and classifier or preceding quantitative expression	ทั้ง, อีก, เพียง
22	DDAQ	Definite determiner, following quantitative expression	พอดี, ถ้วน
23	DIAC	Indefinite determiner, following noun; allowing classifier in between	ไหน, อื่น, ต่าง ๆ
24	DIBQ	Indefinite determiner, between noun and classifier or preceding quantitative expression	บาง, ประมาณ, เกือบ
25	DIAQ	Indefinite determiner, following quantitative expression	กว่า, เศษ
26	DCNM	Determiner, cardinal number expression	หนึ่งคน, เสือ 2 ตัว
27	DONM	Determiner, ordinal number expression	ที่หนึ่ง, ที่สอง, ที่สุดท้าย
28	ADVN	Adverb with normal form	เก่ง, เร็ว, ช้า, สม่ำเสมอ
29	ADVI	Adverb with iterative form	เร็ว ๆ, เสมอ ๆ, ช้า ๆ
30	ADVP	Adverb with prefixed form	โดยเร็ว
31	ADVS	Sentential adverb	โดยปกติ, ธรรมดา
32	CNIT	Unit classifier	ตัว, คน, เล่ม
33	CLTV	Collective classifier	คู่, กลุ่ม, ฝูง, เชิง, ทาง, ด้าน, แบบ, รุ่น
34	CMTR	Measurement classifier	กิโลกรัม, แก้ว, ชั่วโมง
35	CFQC	Frequency classifier	ครั้ง, เที่ยว
36	CVBL	Verbal classifier	ม้วน, มัด
37	JCRG	Coordinating conjunction	และ, หรือ, แต่

ตาราง 2.1 (ต่อ)

ลำดับที่	ชนิดของคำ	รายละเอียด	ตัวอย่างคำ
38	JCMP	Comparative conjunction	กว่า, เหมือนกับ, เท่ากับ
39	JSBR	Subordinating conjunction	เพราะว่า, เนื่องจาก, ที่, แม้ว่า, ถ้า
40	RPRE	Preposition	จาก, ละ, ของ, ใต้, บน
41	INT	Interjection	โอ้ย, โอ้, เออ, เอ๋, อ๋อ
42	FIXN	Nominal prefix	การทำงาน, ความสนุกสนาน
43	FIXV	Adverbial prefix	อย่างเร็ว
44	EAFF	Ending for affirmative sentence	จ๊ะ, จ้ะ, ค่ะ, ครับ, นะ, น่า, เถอะ
45	EITT	Ending for interrogative sentence	หรือ, เหรอ, ไหม, มั้ย
46	NEG	Negator	ไม่, มิได้, ไม่ได้, มิ
47	PUNC	Punctuation	(,), “, ,, ฯ

2.3 ขั้นตอนวิธีวินนาว (Winnow Algorithm)

วินนาว คือ โครงข่ายที่มีลักษณะคล้ายกับ Neural Network โดยจะประกอบไปด้วย Node ต่าง ๆ ซึ่งเรียกว่า Specialist ที่เชื่อมต่อกับ Target Node โดยที่ Specialist แต่ละตัวจะมี Weight ที่แตกต่างกันโดยพิจารณาจากคุณลักษณะของตัวเองเมื่อเปรียบเทียบกับคุณลักษณะของ Target Node ดังรูป 2.1[1]



รูป 2.1 โครงข่ายวินนาว[3]

ขั้นตอนวิธีวินนาวเป็นวิธีการเรียนรู้จากชุดข้อมูลตัวอย่างที่มีการกำหนดประเภทของชุดข้อมูลเอาไว้แล้ว โดยที่ข้อมูลแต่ละตัวภายในชุดข้อมูลจะทำการตรวจสอบคุณลักษณะของตัวเองเปรียบเทียบกับคุณลักษณะของชุดข้อมูลตัวอย่างชุดนั้นเพื่อหาค่าน้ำหนักของตัวเอง จากนั้นวินนาวจะทำนายคุณลักษณะของชุดข้อมูลตัวอย่างชุดนั้นโดยพิจารณาจากค่าน้ำหนักรวมของข้อมูลทั้งหมดภายในชุดข้อมูลชุดนั้นเพื่อใช้ปรับค่าน้ำหนักของข้อมูลแต่ละตัวโดยพิจารณาจากคุณลักษณะของชุดข้อมูลตัวอย่างที่วินนาวทำนายออกมา[1] การเรียนรู้ชุดข้อมูลตัวอย่างด้วยขั้นตอนวิธีวินนาวมีขั้นตอนดังต่อไปนี้

(1) Given: $(<x_i, y_i>)_i \in \{+1,-1\}^n \times \{+1,-1\}$.
(2) Initialize the weight vector by $w \leftarrow (1,1,...,1)$.
(3) Repeat the following process for $(i \leftarrow 1,2,3,...)$:
  (3.1) Given example x_i , predict positive if $w \bullet x \geq n$ and negative if $w \bullet x < n$.
  (3.2) Receiving the correct answer $y_i \in \{+1,-1\}$.
  (3.3) If mistake occurs on a positive example then $w_i \leftarrow w_i \bullet 2$ for every $1 \leq i \leq n$ such that $x_i = 1$.
  (3.4) Else if mistake occurs on a negative example then $w_i \leftarrow w_i \bullet (1/2)$ for every $1 \leq i \leq n$ such that $x_i = 1$.[4]

2.3.1 งานวิจัยเกี่ยวกับขั้นตอนวิธีวินนาว

งานวิจัยของ Charoenpornsawat, P., and Sornlertlamvanich, V.[3] ได้นำขั้นตอนวิธีวินนาวมาใช้ในการพิจารณาคำที่อยู่รอบ ๆ ช่องว่างเพื่อหาขอบเขตของประโยคภาษาไทยโดยใช้ประโยคภาษาไทยในคลังข้อมูลออร์คิดจำนวนประมาณ 9,000 ประโยคเป็นชุดข้อมูลฝึกหัดและใช้ประโยคภาษาไทยในคลังข้อมูลออร์คิดจำนวนประมาณ 1,000 ประโยคเป็นชุดข้อมูลในการทดลอง จากการทดลองพบว่าขั้นตอนวิธีวินนาวสามารถหาขอบเขตของประโยคภาษาไทยได้ถูกต้องประมาณ 89%

2.4 การจัดทำเอกสารให้อยู่ในรูปของ LaTeX

LaTeX คือ รูปแบบการจัดทำเอกสารรูปแบบหนึ่งที่ให้ผู้ใช้ออกแบบโครงสร้างและเนื้อหาของเอกสารที่ต้องการจัดทำขึ้นมาเอง เช่น การระบุชื่อเรื่อง การระบุชื่อหัวเรื่อง การระบุเนื้อหาของแต่ละหัวเรื่อง เป็นต้น ดังตัวอย่างต่อไปนี้

```
\title{\thai รายงานผลการดำเนินงานประจำปี 2550 ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และ
คอมพิวเตอร์แห่งชาติ}

\section{\thai การวิจัยเชิงบูรณาการเพื่อตอบสนองความต้องการของคลัสเตอร์อุตสาหกรรม}

{\thai ในปีงบประมาณ 2550 เนคเทค ได้ดำเนินโครงการและให้ทุนสนับสนุน โครงการวิจัย
พัฒนาและวิศวกรรม ให้เป็นการสนับสนุนในลักษณะของเครือข่ายวิสาหกิจ (Industrial
Clustering) โดยอาศัยความร่วมมือจากทั้งภาครัฐและภาคเอกชน ในการเป็นผู้นำในการวิจัย
พัฒนาและวิศวกรรม ไปจนถึงการถ่ายทอดเทคโนโลยี โดยมีบทบาทหลักในการเน้นการทำงาน
วิจัยเชิงบูรณาการ เพื่อตอบสนองความต้องการของคลัสเตอร์อุตสาหกรรมด้านซอฟต์แวร์
ไมโครชิป และอิเล็กทรอนิกส์ สะท้อนโจทย์ของแต่ละคลัสเตอร์เป็นผลงานที่สำคัญดังนี้}
```

2.5 โปรแกรม Swath

โปรแกรม Swath (Smart Word Analysis for THai) คือ โปรแกรมที่ใช้ในการตัดคำหรือข้อความที่เป็นภาษาไทยจาก Text File, LaTeX File, RTF file หรือ HTML File โดยมีวิธีการตัดคำให้เลือกใช้ 3 วิธี คือ Longest Matching Algorithm, Maximal Matching Algorithm หรือ Bigram Model[7]

2.5.1 Longest Matching Algorithm

การตัดคำด้วยวิธีนี้จะเริ่มจากการจัดเก็บคำต่าง ๆ ไว้ในพจนานุกรม จากนั้นทำการตรวจสอบข้อความหรือประโยคที่ต้องการตัดคำจากซ้ายไปขวาเปรียบเทียบกับคำที่เก็บไว้ในพจนานุกรม ในกรณีที่พบคำในพจนานุกรมมากกว่า 1 คำให้ทำการเลือกคำจากข้อความหรือประโยคนั้นที่มีความยาวมากที่สุด จากนั้นให้ตัดคำที่ถูกเลือกแล้วออกจากข้อความหรือประโยคนั้นแล้วทำการตรวจสอบต่อจนครบทั้งข้อความหรือประโยคนั้น ในกรณีที่เลือกคำที่มีความยาวมากที่สุดแล้วทำให้พบคำที่ไม่มีในพจนานุกรมก็จะทำการย้อนกลับไปตัดคำก่อนหน้าอีกครั้งโดยเลือกคำที่มีความยาวรองลงมาแทน ตัวอย่างการตัดคำด้วยวิธีนี้แสดงดังตาราง 2.2[6]

ตาราง 2.2 ตัวอย่างการตัดคำด้วยขั้นตอนวิธี Longest Matching[1]

ประโยค	คำที่พบในพจนานุกรม	คำที่ถูกตัดออกมา
โคนมนอนบนกองหญ้า	โค, โคน	โคน (เลือกคำที่มีความยาวมากที่สุด)
มนอนบนกองหญ้า	-	(ย้อนกลับไปตัดคำก่อนหน้าอีกครั้ง)
โคนมนอนบนกองหญ้า	โค, โคน	โค (เลือกคำที่มีความยาวรองลงมา)
นมนอนบนกองหญ้า	นม	นม
นอนบนกองหญ้า	นอ, นอน	นอน
บนกองหญ้า	บน	บน
กองหญ้า	กอ, กอง	กอง
หญ้า	หญ้า	หญ้า

จากตาราง 2.2 ประโยคที่นำมาตัดคำคือ “โคนมนอนบนกองหญ้า” ซึ่งการตัดคำด้วยวิธีนี้จะตัดคำออกมาเป็น “โค นม นอน บน กอง หญ้า”

2.5.2 Maximal Matching Algorithm

การตัดคำด้วยวิธีนี้จะเริ่มจากการจัดเก็บคำต่าง ๆ ไว้ในพจนานุกรม จากนั้นทำการตรวจสอบข้อความหรือประโยคที่ต้องการตัดคำจากซ้ายไปขวาเปรียบเทียบกับคำที่เก็บไว้ในพจนานุกรม แล้วทำการตัดคำจากข้อความหรือประโยคนั้นในทุก ๆ กรณีที่สามารถเกิดขึ้นได้ จากนั้นให้ทำการเลือกข้อความหรือประโยคที่ถูกตัดคำออกมาที่มีจำนวนคำน้อยที่สุด ในกรณีที่มีจำนวนคำเท่ากันจะนำวิธีการตัดคำด้วย Longest Matching Algorithm เข้ามาช่วย[6] ตัวอย่างการตัดคำด้วยวิธีนี้แสดงดังตาราง 2.3

ตาราง 2.3 ตัวอย่างการตัดคำด้วยขั้นตอนวิธี Maximal Matching

ประโยค	การตัดคำที่เป็นไปได้	การตัดคำที่ถูกเลือก
ไปหามเหสี	ไป หาม เห สี	ไป หา มเหสี (จำนวนคำน้อยกว่า)
	ไป หา มเหสี	
ฉันนั่งตากลม	ฉัน นั่ง ตาก ลม	ฉัน นั่ง ตาก ลม (จำนวนคำเท่ากัน)
	ฉัน นั่ง ตา กลม	

2.5.3 Bigram Model

การตัดคำด้วยวิธีนี้แบ่งออกเป็น 2 รูปแบบ คือ การตัดคำโดยไม่ระบุชนิดของคำและการตัดคำโดยระบุชนิดของคำ

2.5.3.1 การตัดคำโดยไม่ระบุชนิดของคำ

การตัดคำรูปแบบนี้เป็นการตัดคำโดยพิจารณาจากความน่าจะเป็นของประโยคที่ถูกตัดคำแล้วในคลังข้อมูล ตามสมการ 2.9

$$P(W) = \prod_{i=1}^{n} P(w_{i,n}) = \prod_{i=1}^{n} P(w_i \mid w_{i-1}) \quad \text{สมการ 2.9}$$

โดยที่ $W = w_1 \ldots w_n$ คือ ประโยคที่ถูกตัดคำแล้วซึ่งจะประกอบไปด้วยคำต่าง ๆ ตั้งแต่ w_1 จนถึง w_n, $P(w_{i,n})$ คือ ความน่าจะเป็นของ W ในคลังข้อมูล และ $P(w_i \mid w_{i-1})$ คือ ความน่าจะเป็นของ w_i ซึ่งมีค่าขึ้นอยู่กับ w_{i-1} เท่านั้น

2.5.3.2 การตัดคำโดยระบุชนิดของคำ

การตัดคำรูปแบบนี้เป็นการตัดคำโดยพิจารณาจากความน่าจะเป็นของประโยคที่ถูกตัดคำและระบุชนิดของคำแล้วในคลังข้อมูล ตามสมการ 2.10

$$P(W_i) = \sum_{T} P(W_i, T_i) = \sum_{T} \prod_{i} P(t_i \mid t_{i-1}) \times P(w_i \mid t_i) \quad \text{สมการ 2.10}$$

โดยที่ $W_i = w_1 \ldots w_n$ คือ ประโยคที่ถูกตัดคำแล้วซึ่งจะประกอบไปด้วยคำต่าง ๆ ตั้งแต่ w_1 จนถึง w_n, $T_i = t_1 \ldots t_n$ คือ ชนิดของคำของ w_1 จนถึง w_n, $P(W_i, T_i)$ คือ ความน่าจะเป็นของ W_i ที่มี T_i เป็นชนิดของคำ, $P(t_i \mid t_{i-1})$ คือ ความน่าจะเป็นของ t_i ซึ่งมีค่าขึ้นอยู่กับ t_{i-1} เท่านั้น และ $P(w_i \mid t_i)$ คือ ความน่าจะเป็นของ w_i ซึ่งมีค่าขึ้นอยู่กับ t_i เท่านั้น