ระบบสืบค้นสำหรับประโยคภาษาไทย ใช้วิธีถ่วงน้ำหนักช่วงข้อความที่สั้นที่สุด

สมหมาย ขันทอง* พยุง มีสัจ** และชูชาติ หฤไชยะศักดิ์***

บทคัดย่อ

ระบบสืบค้นเป็นระบบคำถาม-คำตอบที่ให้ความสะดวกแก่ผู้ใช้ สำหรับการเข้าถึงสารสนเทศที่เป็นข้อความในเอกสารอิเล็กทรอนิกส์ด้วยการรับคำขอจากผู้ใช้ในรูปแบบประโยคคำถามที่เป็นภาษาธรรมชาติ และได้รับผลลัพธ์เป็นคำตอบที่กระชับรวบรัด ซึ่งการค้นคืนเอกสารและการสกัดข้อความสั้นเป็นส่วนที่มีผลกระทบต่อประสิทธิภาพของการค้นหาคำตอบ ในงานวิจัยนี้จึงได้ใช้วิธีถ่วงน้ำหนักช่วงข้อความที่สั้นที่สุดมาเพิ่มประสิทธิภาพการจัดลำดับเอกสารที่ถูกค้นคืนและการสกัดข้อความสั้นที่น่าจะมีคำตอบที่ต้องการอยู่ เพื่อลดความยุ่งยากในการวิเคราะห์เชิงภาษาศาสตร์ โดยเฉพาะเมื่อใช้กับข้อความภาษาไทยที่มีรูปแบบและโครงสร้างทางภาษาที่ไม่แน่นอน ผลการประเมินประสิทธิภาพได้ค่าลำดับคำตอบส่วนกลับเฉลี่ย (MRAR) เท่ากับ 0.81 และเพื่อเพิ่มประสิทธิภาพการเลือกคำตอบจึงได้พัฒนาระบบการแบ่งประโยคภาษาไทย โดยใช้เทคนิคกลไกการเรียนรู้ด้วยวิธีซัพพอร์ตเวกเตอร์แมชชีน ซึ่งจากการทดลองกับคลังข้อมูลออร์คิดด้วยวิธี 10-fold cross validation พบว่าสามารถจำแนกประเภทช่องว่างและแบ่งประโยคภาษาไทยถูกต้องเฉลี่ย 92.46% และ 85.92% ตามลำดับ

คำสำคัญ : ระบบคำถาม-คำตอบ การแบ่งประโยคภาษาไทย ระบบค้นคืนสารสนเทศ ซัพพอร์ตเวกเตอร์แมชชีน

1. บทนำ

ปัจจุบันระบบการค้นคืนสารสนเทศ (Information Retrieval System) ใช้วิธีการนำคำสั้น ๆ หรือคำสำคัญไปเปรียบเทียบกับแหล่งข้อมูล เพื่อให้ได้ผลลัพธ์ตรงกับคำที่ต้องการค้นหา แต่ผลลัพธ์ที่ได้ยังไม่เพียงพอต่อความต้องการของผู้ใช้ที่ต้องการค้นหาคำตอบเฉพาะเรื่อง จากคำถามง่าย ๆ เช่น "*ใครเป็นประธานาธิบดีของประเทศสหรัฐอเมริกา*" ซึ่งระบบจะไม่ตอบคำถามโดยตรง แต่บอกว่าจะหาคำตอบได้จากที่ไหน โดยแสดงรายการเอกสารที่อาจจะมีคำตอบอยู่ แล้วผู้ใช้กลั่นกรองเพื่อหาคำตอบด้วยตนเองอีกที แสดงให้เห็นว่าระบบค้นคืนสารสนเทศเพียงอย่างเดียวไม่สามารถตอบสนองความต้องการของผู้ใช้ได้ ระบบคำถาม-คำตอบ (Question-Answering System) จึงได้รับการพัฒนาขึ้นเพื่อตอบสนองต่อความต้องการสืบค้นข้อมูลในรูปแบบการถาม-ตอบ และเพิ่มประสิทธิภาพของระบบค้นคืนสารสนเทศ โดยรับคำขอ (Query) ในรูปแบบภาษาธรรมชาติและนำเสนอผลลัพธ์เป็นคำตอบสั้นๆ แทนข้อความทั้งเอกสารที่มีคำตอบเหล่านั้นอยู่

จากการตรวจเอกสารพบว่างานวิจัยที่เกี่ยวข้องกับการพัฒนาระบบคำถาม-คำตอบ ส่วนมากใช้กับข้อความภาษาอังกฤษ โดยเป็นของกลุ่มนักวิจัยภายใต้การประชุมวิชาการ Text Retrieval Evaluation Conference (TREC) ซึ่งมีสถาปัตยกรรมทั่วไปของระบบประกอบด้วย 4 ส่วนหลัก คือ การประมวลผลคำถาม การประมวลผลเอกสาร การประมวลผลคำตอบ และการสร้างคำตอบ โดยมีแนวทางการพัฒนาระบบแบ่งออกเป็น 2 กลุ่มหลัก คือการพัฒนาที่มุ่งเน้นด้านฐานความรู้ [1]-[3] และการพัฒนาที่มุ่งเน้นด้านข้อมูล [4]-[6] กลุ่มแรกเป็นกลุ่มที่มุ่งเน้นด้านฐานความรู้ จะเน้นการนำเครื่องมือวิเคราะห์ทางภาษาศาสตร์ เช่น การแจงประโยค (Parsing) การวิเคราะห์ประโยค (Syntactic Analysis) การวิเคราะห์ความหมายของคำ (Semantic Analysis) และฐานความรู้ทางภาษาศาสตร์ (Linguistics Knowledge) มาใช้ในการประมวลผลทั้งคำถามและคำตอบเพื่อวิเคราะห์ความหมายอย่างละเอียด ซึ่งแนวทางนี้มีข้อจำกัดในเรื่องของเครื่องมือที่จะนำมาใช้ เนื่องจากเครื่องมือเหล่านี้ไม่ได้มีพร้อมใช้กับทุกภาษา ระบบที่พัฒนาขึ้นสามารถใช้ได้กับภาษาใดภาษาหนึ่งเท่านั้น และประสิทธิภาพของระบบจะขึ้นอยู่กับความครอบคลุมของฐานความรู้ทางภาษาศาสตร์

กลุ่มที่สองเป็นกลุ่มที่มุ่งเน้นด้านข้อมูลจะใช้ประโยชน์จากข้อมูลที่มีปริมาณมากมาใช้แทนข้อมูลทางภาษาศาสตร์ที่มีข้อจำกัดทางด้านภาษา โดยไม่ได้มุ่งเน้นที่จะทำความเข้าใจ

* *นักศึกษา ระดับปริญญาโท คณะเทคโนโลยีสารสนเทศ สจพ.*

** *อาจารย์ ระดับ 7 ภาควิชาครุศาสตร์ไฟฟ้า คณะครุศาสตร์อุตสาหกรรม สจพ. และ รองคณบดีฝ่ายบริหาร คณะเทคโนโลยีสารสนเทศ สจพ.*

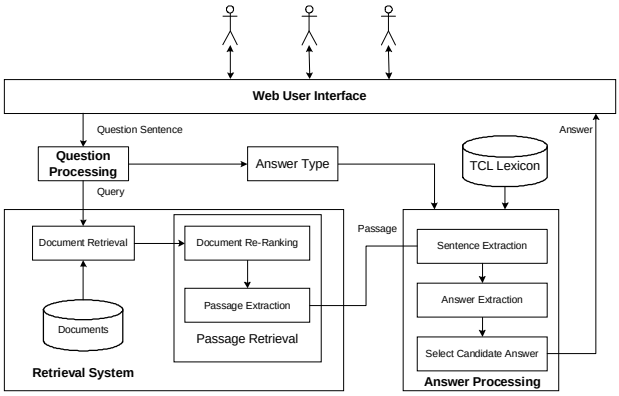
*** *นักวิจัยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ อาจารย์พิเศษ คณะเทคโนโลยีสารสนเทศ สจพ.*

ถึงความหมายแท้จริงของทั้งคำตอบและคำถามอย่างลึกซึ้ง แต่เป็นการวิเคราะห์แบบหยาบ และใช้เทคนิควิธีกลไกการเรียนรู้ (Machine Learning) หรือวิธีการทางสถิติมาใช้ในการวิเคราะห์ข้อมูล แนวทางการพัฒนานี้จึงมีข้อจำกัดน้อยกว่าแนวทางแรก และสามารถนำไปประยุกต์ใช้กับภาษาอื่น ๆ ได้ แต่ความถูกต้องและความครอบคลุมของคำตอบน้อยกว่าวิธีการแรก และประสิทธิภาพของระบบก็จะขึ้นอยู่กับปริมาณข้อมูลที่สามารถนำมาใช้ได้ รวมทั้งเทคนิควิธีการที่ใช้ในขบวนการวิเคราะห์ข้อมูลด้วย

ดังนั้นในงานวิจัยนี้จึงได้พัฒนาตามแนวทางที่มุ่งเน้นด้านข้อมูล ซึ่งเป็นการวิเคราะห์ทั้งคำถามและคำตอบแบบหยาบ โดยมีส่วนประกอบหลัก คือ การประมวลผลคำถาม การค้นคืนเอกสาร การสกัดข้อความสั้น และการประมวลผลคำตอบ โดยเฉพาะการค้นคืนเอกสารและสกัดข้อความสั้นเป็นส่วนที่มีผลกระทบต่อประสิทธิภาพของการค้นหาคำตอบ ในงานวิจัยนี้จึงได้ใช้วิธีถ่วงน้ำหนักช่วงข้อความที่สั้นที่สุด [7] มาเพิ่มประสิทธิภาพการจัดลำดับเอกสารและการสกัดข้อความสั้นที่น่าจะมีคำตอบที่ต้องการอยู่ เพื่อลดข้อจำกัดของภาษาไทยและให้มีความเหมาะสมต่อการนำมาใช้กับข้อความภาษาไทยที่มีรูปแบบทางภาษาไม่แน่นอน เช่นเดียวกับงานวิจัยของ [8] ที่ใช้วิธีการจัดหมวดหมู่เอกสารเข้ามาใช้ในระบบ เพื่อลดความยุ่งยากในการวิเคราะห์เชิงภาษาศาสตร์ โดยระบบที่พัฒนาขึ้นนี้มุ่งเน้นการค้นหาข้อความในเอกสารโดยดูจากความใกล้ชิดกันของคิวรีเทอมที่ปรากฏในเอกสาร ซึ่งเรียกข้อความนี้ว่าช่วงข้อความที่สั้นที่สุด และกำหนดข้อความนี้เป็นข้อความสั้น (Passage) นำไปสกัดคำตอบให้กับคำถามต่อไป โดยการค้นหาช่วงข้อความที่สั้นที่สุดนั้นได้ประยุกต์ใช้อัลกอริทึม แพลน-สวีพ (Plane-Sweep) ที่มีความเร็วของการค้นหาอยู่ที่ $O(n \log n)$ เมื่อ n เป็นจำนวนตำแหน่งที่ปรากฏในเอกสารของเทอมจำนวน k เทอม [9] เพื่อให้การค้นหามีความรวดเร็วยิ่งขึ้น และในส่วนของการประมวลผลคำตอบใช้วิธีการดูความใกล้ชิดกันระหว่างคิวรีเทอมและคำตอบที่ปรากฏในเอกสาร โดยวัดจากระยะห่างระหว่างตำแหน่งที่ปรากฏและระยะห่างระหว่างประโยคในข้อความ ซึ่งการระบุขอบเขตของแต่ละประโยคนั้นในงานวิจัยของ [10] ใช้แบบจำลองไตรแกรมกำกับหน้าที่ของคำ และใช้วิธีกลไกเรียนรู้วินโนว์ (Winnow) [11] เพื่อแยกแยะช่องว่างว่าเป็นตัวแบ่งประโยคหรือไม่ โดยใช้บริบทรอบ ๆ ช่องว่างเป็นคุณลักษณะสำหรับการเรียนรู้ ในงานวิจัยนี้จึงใช้แนวความคิดเดียวกัน และได้นำเสนอวิธีกลไกการเรียนรู้ด้วยวิธีการซัพพอร์ตเวกเตอร์แมชชีนสำหรับการเรียนรู้เพื่อแยกแยะประเภทของช่องว่าง เนื่องจากเป็นวิธีการที่เหมาะสำหรับการแก้ปัญหาการรู้จำรูปแบบข้อมูลที่ต้องการแยกแยะออกเป็นสองกลุ่ม [12]

2. วิธีการดำเนินงาน

ขอบเขตของงานวิจัยนี้ครอบคลุมกระบวนการทั้งหมดในการพัฒนาระบบคำถาม-คำตอบ โดยได้แบ่งส่วนประกอบของระบบออกเป็น 6 ส่วนหลัก ดังแสดงในภาพที่ 1 ซึ่งมีรายละเอียดดังต่อไปนี้



ภาพที่ 1 ภาพรวมของระบบคำถาม-คำตอบ

2.1 การประมวลผลคำถาม

การประมวลผลคำถามเป็นการวิเคราะห์เพื่อทำความเข้าใจคำถามของผู้ใช้ ว่าคำถามนั้นต้องการคำตอบอะไร ด้วยการแบ่งประเภทของคำถามตามประเภทของคำตอบที่คาดหวัง โดยนำข้อความประโยคคำถามมาตัดคำให้อยู่ในรูปเวกเตอร์ของคำและใช้วิธีสร้างรูปแบบ (Regular Expression) ของแต่ละประเภท ซึ่งดูจากคำแสดงคำถามเป็นหลัก แต่เนื่องจากคำแสดงคำถามบางคำมีความกำกวมในการระบุประเภท เช่น “อะไร” สามารถระบุได้หลายประเภท ดังนั้นจึงต้องพิจารณาคำหรือวลีอื่นๆ ประกอบด้วย ตัวอย่างเช่น “แหล่งปิโตรเลียมแห่งแรกของไทยพบที่จังหวัดอะไร” จากประโยคนี้ “จังหวัด” เป็นคำที่ระบุถึงสถานที่ คำถามนี้จึงจัดอยู่ในประเภทเกี่ยวกับสถานที่ โดยในงานวิจัยนี้ได้ใช้ทดลองกับคำถาม ซึ่งแบ่งตามประเภทคำตอบ ดังแสดงในภาพที่ 2

การสร้างคำขอจากเวกเตอร์ของคำที่ถูกขจัดคำแสดงคำถามและคำไม่สำคัญหรือคำหยุด (Stopword) โดยใช้ชุด

คำหยุดจากงานวิจัยของ [13] เพื่อนำไปใช้ในขั้นตอนการค้นคืนเอกสาร

HUMAN	NUMERIC
- group	- count
- individual	- distance
LOCATION	- prices
- continent	- ranks
- cities	- other
- countries	- area
- other	- capacity
DATE	- speed
- date	- temperature
- month	- size
- year	- weight
- time expression	
- interval	

ภาพที่ 2 ประเภทของคำตอบ

2.2 การค้นคืนเอกสาร

การค้นคืนเอกสารเป็นการค้นหาเอกสารที่มีความสัมพันธ์กับคำขอ ซึ่งในงานวิจัยนี้ได้ใช้ชุดโปรแกรมเอพีไอไลบรารี่ใน Lucene [14] สำหรับสร้างโปรแกรมค้นคืนเอกสาร และใช้วิธีการวัดความเหมือนระหว่างคำขอและเอกสารด้วยตัวแบบปริภูมิเวกเตอร์ (Vector Space Model) [15] ซึ่งการถ่วงน้ำหนักแต่ละเทอมใช้วิธีที่เรียกว่า TF-IDF เป็นค่าผลคูณระหว่าง TF (Term Frequency) และ IDF (Inverse Document Frequency) คำนวณได้ดังสมการที่ (1)

$$TF \times IDF = freq(t_j, d) \times \log\left(\frac{N}{n_j}\right) \qquad (1)$$

โดยที่ d : เอกสาร, $freq(t_j, d)$: ความถี่ของเทอม j ในเอกสาร d , N : จำนวนเอกสารทั้งหมดในคลังเอกสาร, n_j เป็นจำนวนเอกสารที่มีเทอมปรากฏอยู่

การวัดความเหมือนของเอกสารใช้วิธีวัดความเหมือนเชิงมุม (Cosine Similarity) ดังสมการที่ (2) [16]

$$\cos(\vec{q}, \vec{d}) = \frac{\vec{q}\,\vec{d}}{|\vec{q}||\vec{d}|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2 \sum_{i=1}^{n} w_{i,q}^2}} \qquad (2)$$

โดยที่ $\vec{q}$ เป็นคำขอ ประกอบด้วยเทอมย่อยๆ $\vec{q} = (q_1, q_2, ..., q_n)$, $\vec{d}$ เป็นเอกสารประกอบด้วยเอกสารหลายฉบับ $\vec{d} = (d_1, d_2, ..., d_n)$, $w_{i,j}$ เป็นค่าถ่วงน้ำหนักของเทอม i ในเอกสาร j และ $w_{i,q}$ เป็นค่าถ่วงน้ำหนักของเทอม i ในคำขอ q

เนื่องจากผลลัพธ์ของระบบคำถาม-คำตอบ จะเป็นคำตอบโดยตรงหรือข้อความสั้น ซึ่งคำตอบเหล่านี้เป็นเพียงส่วนใดส่วนหนึ่งของเอกสารไม่ได้อยู่ทั้งเอกสาร ดังนั้นจึงต้องจัดลำดับเอกสารใหม่ ซึ่งในงานวิจัยนี้ใช้วิธีถ่วงน้ำหนักช่วงข้อความที่สั้นที่สุด [7] โดยพิจารณาจากช่วงข้อความที่สั้นที่สุดที่มีคิวรีเทอมปรากฏอยู่มากที่สุด ด้วยวิธีการวัดระยะห่างของแต่ละคิวรี่เทอมที่ปรากฏในเอกสาร ซึ่งระยะห่างนี้จะนับจำนวนคำที่คั่นระหว่างเทอม และประยุกต์ใช้อัลกอริทึมแพลน-สวีพ (Plane-Sweep) [9] เพื่อหาช่วงข้อความที่สั้นที่สุด และคำนวณค่าความเหมือนของเอกสารเพื่อจัดลำดับเอกสารแสดงดังสมการที่ (3) [7]

$$RSV(q,d) = \lambda RSV_n(q,d) + (1-\lambda) \left(\frac{|q \mathbf{I} d|}{1 + \max(mms) - \min(mms)}\right)^{\alpha} \left(\frac{|q \mathbf{I} d|}{q}\right)^{\beta} \qquad (3)$$

โดย $RSV(q,d)$ เป็นคะแนนความเหมือนของเอกสารที่วัดทั้งเอกสารที่ผ่านการนอร์มอนไลซ์แล้ว

เทอม $\left(\frac{|q \mathbf{I} d|}{1 + \max(mms) - \min(mms)}\right)^{\alpha}$ เป็นอัตราส่วนของขนาดช่องข้อความ (Span Size Ratio)

เทอม $\left(\frac{|q \mathbf{I} d|}{q}\right)^{\beta}$ เป็นอัตราส่วนของเทอมในเอกสารที่ตรงกับคิวรีเทอม (Matching Term Ratio)

$\min(mms)$ เป็นตำแหน่งซ้ายสุดของช่วงข้อความ

$\max(mms)$ เป็นตำแหน่งขวาสุดของช่วงข้อความ

$\lambda = 0.4,\ \alpha = 1/8,\ \beta = 1$

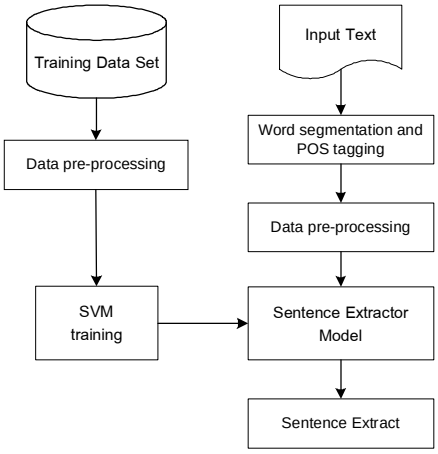
2.3 การสกัดข้อความสั้น

ชุดเอกสารที่ถูกจัดลำดับเอกสารแล้ว จะเลือกเอกสารมาจำนวนหนึ่ง ซึ่งมีคะแนนห่างจากเอกสารที่มีคะแนนสูงสุดไม่เกิน 25% นำมาสกัดข้อความสั้น โดยกำหนดขอบเขตข้อความสั้นในระดับย่อหน้า ซึ่งพิจารณาย่อหน้าที่มีช่วง

ข้อความที่สั้นที่สุดปรากฏอยู่ หากช่วงข้อความที่สั้นที่สุดคาบเกี่ยวระหว่างสองย่อหน้าให้รวมสองย่อหน้านั้นเป็นข้อความสั้นเดียวกัน และหากย่อหน้าไหนมีจำนวนคำน้อยกว่า 250 คำ ให้ไปรวมกับย่อหน้าข้างเคียง

2.4 การแบ่งประโยคข้อความ

แนวความคิดในการแบ่งประโยคข้อความภาษาไทยนั้นมีลักษณะเดียวกับงานวิจัยของ [10] และ [11] ซึ่งใช้วิธีการแยกแยะช่องว่างที่ปรากฏบนข้อความออกเป็น 2 กลุ่มด้วยกัน คือ ช่องว่างที่เป็นตัวแบ่งประโยค และไม่ใช่เป็นตัวแบ่งประโยค โดยใช้บริบทรอบ ๆ ช่องว่างมาเป็นคุณลักษณะสำหรับใช้ในการเรียนรู้ ซึ่งการแยกแยะช่องว่างดังกล่าวในงานวิจัยนี้ใช้เทคนิคกลไกการเรียนรู้ด้วยวิธีการซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machines: SVMs) โดยใช้เพคเกจโปรแกรม Tiny SVM [17] สำหรับสร้างแบบจำลองจำแนกประเภทข้อมูล และใช้โปรแกรม Yamcha [18] สำหรับแปลงข้อมูลเพื่อใช้สำหรับการฝึกฝนระบบ โดยมีขั้นตอนการทำงานแสดงดังภาพที่ 3



ภาพที่ 3 ขั้นตอนการทำงานของการแบ่งประโยค

การสร้างแบบจำลองสำหรับการแบ่งประโยคข้อความการทำงานเริ่มจากนำชุดข้อมูลฝึกฝน ซึ่งในงานวิจัยนี้ได้เลือกใช้คลังข้อมูลออร์คิด [19] มาผ่านกระบวนการจัดเตรียมข้อมูลเพื่อสร้างคุณลักษณะให้กับช่องว่างที่จะใช้ในการพิจารณาทั้งหมดในคลังข้อมูล ซึ่งมีจำนวน 14 ชุด โดย 12 ชุดแรกเป็นคำและหน้าที่ของคำที่อยู่หน้าและหลังช่องว่างจำนวน 3 คำ อีก 2 ชุดหลังเป็นจำนวนคำที่อยู่หน้าช่องว่างและหลังช่องว่าง จากนั้นกำหนดเฉลย (Class Label) ให้กับข้อมูลแต่ละชุด ซึ่งในที่นี้คำตอบว่าช่องว่างนั้นเป็นช่องว่างแบ่งประโยคหรือไม่ และนำชุดข้อมูลเข้าสู่กระบวนการเรียนรู้เพื่อสร้างแบบจำลองสำหรับการแบ่งประโยคข้อความการใช้แบบจำลองที่สร้างขึ้นเพื่อแบ่งประโยคข้อความจากเอกสารใหม่ การทำงานเริ่มจากนำเอกสารมาผ่านกระบวนการตัดคำและกำกับหน้าที่ของคำโดยใช้โปรแกรมตัดคำ SWATH [20] และจัดเตรียมข้อมูลเพื่อสร้างคุณลักษณะให้กับช่องว่างที่ปรากฏในเอกสารเข้าสู่แบบจำลองสำหรับการแบ่งประโยคข้อความที่สร้างจากขั้นตอนการฝึกฝนระบบ ผลลัพธ์ที่ได้คือ ข้อความที่ผ่านการแบ่งประโยคโดยใช้ช่องว่างที่กำหนดขึ้นจากแบบจำลอง

2.5 การสกัดคำตอบ

การสกัดคำตอบเป็นการค้นหาคำตอบที่อยู่ในข้อความสั้น ซึ่งการสกัดคำตอบนี้จะสกัดตามประเภทของคำตอบที่กำหนดไว้ในขั้นตอนการประมวลคำถาม โดยใช้รูปแบบคำตอบของแต่ละประเภทคำตอบ และพจนานุกรมเชิงความหมายทีซีแอล [21] สำหรับคำตอบประเภทบุคคลและองค์กร ซึ่งจัดเก็บคำศัพท์แบ่งออกเป็นกลุ่มๆ ตามแนวความคิด (Concept) และมีความสัมพันธ์กันเป็นลำดับขั้น ตัวอย่างกลุ่มแนวความคิด person เช่น “นายกรัฐมนตรี”, “ครู”, “คนไทย”, “ช่างไม้” เป็นต้น ส่วนรูปแบบคำตอบนั้นสร้างจากบริบทบ่งบอก [22] ของคำตอบแต่ละประเภท เช่น “[๐-๙0-9]+[๐-๙0-9|\,|\.|\s?\-\s?]+(\s+)?(ล้าน|แสน|หมื่น|พัน)?(กิโลเมตร|เมตร|หลา|ไมล์|ไมล์ทะเล)” สำหรับคำตอบประเภทตัวเลขบอกระยะทาง

2.6 การจัดลำดับและเลือกคำตอบ

จากชุดคำตอบคู่แข่ง (Candidate Answer) ที่ได้จากขั้นตอนการสกัดคำตอบ นำมาเลือกคำตอบที่น่าจะเป็นคำตอบที่ถูกต้องที่สุด โดยใช้บริบทรอบ ๆ คำตอบเป็นเกณฑ์ในการพิจารณา ซึ่งคำนวณได้จากสมการที่ (4)

$$Context\_scored_{(Q,A)} = \sum_{q_i \in Q} proximity_{d(q_i,A)} \times span\_wieght_{d(q_i,A)} \quad (4)$$

จากสมการที่ (4) $Context\_score$ เป็นคะแนนที่คำนวณจากความใกล้ชิดกันระหว่างคำตอบ A และคิวรีเทอม q_i ที่ q_i เป็นสมาชิกของ Q และค่าถ่วงน้ำหนักของช่วงข้อความที่สั้นที่สุด $span\_wieght_{d(q_i,A)}$ ที่มีคำตอบ A และคิวรีเทอม q_i ปรากฏอยู่ ซึ่งคำนวณได้จากสมการ (3) แต่ขนาดช่วงข้อความ

จะขยายจุดเริ่มต้นของประโยคเป็นขอบด้านซ้าย $(\min(mms))$ และจุดสิ้นสุดประโยคเป็นขอบด้านขวาของช่วงข้อความ $(\max(mms))$

$$proximity_{d(q_i,A)} = \begin{cases} \dfrac{idf_i}{dist_{d(q_i,A)}} & \text{if } q_i \text{ and } A \text{ exists} \\ & \text{in the passage } d \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

จากสมการที่(5) $proximity_{d(q_i,A)}$ คำนวณจากความใกล้ชิดกันระหว่างคำตอบและคิวรีเทอมแต่ละเทอม โดยพิจารณาจากค่า *idf* ของแต่ละคิวรีเทอมและระยะห่างระหว่างคำตอบและคิวรีเทอม

$$dist_{d(q_i,A)} = sent\_dist_{d(q_i,A)} \times \log\left(pos\_dist_{d(q_i,A)} + 1\right) \quad (6)$$

จากสมการที่ (6) $dist_{d(q_i,A)}$ คำนวณระยะห่างระหว่างคำตอบและคิวรีเทอมโดยพิจารณาระยะห่างระหว่างประโยค $sent\_dist_{d(q_i,A)}$ และระยะห่างระหว่างตำแหน่งที่ปรากฏในข้อความ $pos\_dist_{d(q_i,A)}$ ซึ่งการวัดระยะห่างระหว่างตำแหน่งนี้จะใช้วิธีการนับจำนวนคำที่อยู่ห่างกันในข้อความ โดยข้อความนี้จะต้องทำการขจัดคำหยุดทิ้งไปก่อน ซึ่งคำตอบที่มีคะแนนสูงสุดจะถูกเลือกให้กับคำถาม และเรียงลำดับคำตอบตามค่าคะแนน

3. ผลการทดลองและวิจารณ์

3.1 การแบ่งประโยคภาษาไทย

การทดสอบการแบ่งประโยคภาษาไทยในงานวิจัยนี้ใช้ข้อมูลจากคลังข้อมูลออร์คิด โดยทดสอบแบบจำลองที่ได้จากการฝึกฝนระบบด้วยวิธี 10-fold Cross Validation ทำการวัดเปอร์เซ็นต์ความถูกต้องของการแบ่งประโยค (Break-Correct) เปอร์เซ็นต์ความถูกต้องของการจำแนกประเภทของช่องว่าง (Space-Correct) และเปอร์เซ็นต์ความผิดพลาดของการแบ่งประโยค (False-Break) ซึ่งคำนวณได้จากสมการที่ (7), (8) และ (9) ตามลำดับ [10]-[11] ผลลัพธ์แสดงในตารางที่ 1

$$Break-correct = \left(CB / RB\right) \times 100\% \quad (7)$$

$$Space-correct = \left(CS / RS\right) \times 100\% \quad (8)$$

$$False-break = \left(FB / RS\right) \times 100\% \quad (9)$$

โดย CB เป็นจำนวนความถูกต้องของการจำแนกประเภทช่องว่างที่เป็นตัวแบ่งประโยค

FB เป็นจำนวนความผิดพลาดของการจำแนกประเภทช่องว่างที่เป็นตัวแบ่งประโยค

CS เป็นจำนวนความถูกต้องของการจำแนกประเภทช่องว่างที่เป็นและไม่เป็นตัวแบ่งประโยค

RS เป็นจำนวนช่องว่างที่เป็นตัวแบ่งประโยคและช่องว่างที่ไม่ใช่ตัวแบ่งประโยค

RB เป็นจำนวนช่องว่างที่เป็นตัวแบ่งประโยค

ตารางที่ 1 ผลการทดลองการแบ่งประโยคภาษาไทย

Test set	Number of spaces		%Space-correct	%Break-correct	%False-break
	SBS	NSBS			
1	2071	4996	92.22	84.36	4.58
2	1801	5349	93.36	87.01	3.27
3	2096	5034	91.99	85.69	4.21
4	1962	5112	92.10	86.29	3.80
5	1859	5226	92.80	87.14	3.37
6	2100	4908	91.34	85.33	4.39
7	1966	5147	91.47	81.74	5.05
8	2105	4916	92.47	84.51	4.64
9	2311	4768	91.59	81.91	5.90
10	2314	4606	95.30	91.88	2.72
Average			**92.46**	**85.59**	**4.19**

จากผลการทดลอง เปอร์เซ็นต์ความถูกต้องของการแบ่งประโยคเฉลี่ยมีค่าเท่ากับ 85.59% เปอร์เซ็นต์ความถูกต้องของการจำแนกประเภทของช่องว่างเฉลี่ยมีค่าเท่ากับ 92.46% และเปอร์เซ็นต์ของความผิดพลาดของการแบ่งประโยคเฉลี่ยมีค่าเท่ากับ 4.19% ซึ่งมีความถูกต้องสูงกว่าวิธีแบบจำลองไตรแกรมกำกับหน้าที่คำ [10] และวินโนว์ [11]

3.2 ระบบคำถาม-คำตอบ

ชุดข้อมูลที่ใช้สำหรับการทดสอบในงานวิจัยนี้ ใช้ข้อมูลจากสารานุกรมไทยสำหรับเยาวชนฯ จำนวนทั้งสิ้น 6,472

เอกสาร ขนาด 25.4 เมกะไบต์ และชุดคำถาม 5 ประเภท การวัดประสิทธิภาพของระบบใช้วิธีวัดค่าลำดับคำตอบส่วนกลับเฉลี่ย (Mean Reciprocal Answer Rank: MRAR) ดังสมการที่ (10) [23] โดยป้อนคำถามให้กับระบบที่พัฒนาขึ้น จากนั้นนำผลลัพธ์ที่ 5 อันดับแรก มาวัดประสิทธิภาพ ดังแสดงในตารางที่ 2

$$MRAR = \frac{1}{n}\left(\sum_{i=1}^{n}\frac{1}{r_i}\right) \qquad (10)$$

เมื่อให้ n เป็นจำนวนของคำถาม และ r_i เป็นเลขอันดับที่คำตอบที่ถูกต้องปรากฏอยู่ของคำถามที่

ตารางที่ 2 ผลลัพธ์คำตอบที่ถูกต้องใน 5 อันดับแรก

Rank	#1	#2	#3	#4	#5	Failure
บุคคล	27	4	0	0	0	4
องค์กร	20	3	2	2	1	7
สถานที่	50	9	4	1	1	5
วันเวลา	74	16	5	4	0	1
ตัวเลข	78	13	3	1	0	5
รวม	249	45	14	8	2	22
MRAR	**0.81**					

การวัดประสิทธิภาพของระบบคำตอบ-คำถามที่ได้จากงานวิจัยนี้ โดยได้ทำการทดลองกับชุดคำถาม 5 ประเภท จำนวน 340 คำถาม ซึ่งมีผลลัพธ์ได้ค่า MRAR = 0.81 และอัตราความถูกต้องของคำตอบ (5 อันดับแรก) = 93.53%

จากผลการวัดประสิทธิภาพชุดคำถามประเภทองค์กรและบุคคล มีความถูกต้องน้อยกว่าประเภทอื่น ๆ เนื่องจากการสกัดคำตอบใช้วิธีการสร้างรูปแบบ โดยสร้างจากบริบทบ่งบอก ซึ่งบริบทบ่งบอกเหล่านี้มีขีดความสามารถในการระบุชื่อเฉพาะต่างกัน และมีความกำกวมที่จะระบุได้ว่าเป็นชื่อเฉพาะหรือไม่ ทำให้การสกัดคำตอบประเภทนี้มีความถูกต้องลดลงด้วย สำหรับชุดคำถามประเภทตัวเลขเชิงปริมาณและวันเวลา การใช้รูปแบบคำตอบนั้นมีความถูกต้องสูง เนื่องจากลักษณะรูปแบบของวันที่ เวลา หรือตัวเลขเชิงปริมาณที่ปรากฏในข้อความมีลักษณะตายตัว ทำให้การสกัดคำตอบประเภทนี้มีความถูกต้องสูงตามไปด้วย และสำหรับความถูกต้องของคำตอบในแต่ละอันดับนั้น มีปัจจัยที่มีผลกระทบต่อความถูกต้องจากการแบ่งประโยค หากสามารถแบ่งประโยคได้ถูกต้อง การเลือกคำตอบมีความถูกต้องเพิ่มขึ้นตามไปด้วย

4. สรุป

บทความนี้ได้นำเสนอระบบสืบค้นที่ใช้สำหรับประโยคภาษาไทยโดยใช้วิธีการถ่วงน้ำหนักช่วงข้อความที่สั้นที่สุด ซึ่งระบบมีส่วนประกอบหลัก 6 ส่วน คือ การประมวลผลคำถาม การค้นคืนและจัดลำดับเอกสาร การสกัดข้อความสั้น การแบ่งประโยคข้อความ การสกัดคำตอบ การจัดลำดับและเลือกคำตอบ โดยเฉพาะในส่วนการค้นคืนเอกสารและการสกัดข้อความสั้น เป็นส่วนสำคัญที่มีผลกระทบต่อประสิทธิภาพของการค้นหาคำตอบจากข้อความในเอกสารที่ถูกค้นคืนมาได้ ในงานวิจัยนี้ได้นำวิธีการถ่วงน้ำหนักข้อความที่สั้นที่สุด ซึ่งเป็นการประยุกต์เทคนิคของงานด้านระบบค้นคืนสารสนเทศมาใช้ในระบบ เพื่อลดข้อจำกัดของภาษาไทย ซึ่งผลการทดลองแสดงให้เห็นว่าวิธีการนี้สามารถเพิ่มประสิทธิภาพในส่วนของการค้นคืนเอกสารและการสกัดข้อความสั้นที่ทำให้สามารถค้นหาคำตอบได้ถูกต้องในระดับที่น่าพอใจ โดยที่มีผลกระทบจากข้อจำกัดของภาษาไทยต่อระบบน้อย

การวิจัยในอนาคตควรเพิ่มการขยายคำขอ เพื่อให้สามารถค้นคืนเอกสารได้ครอบคลุมเพิ่มขึ้น และเพิ่มส่วนการรู้จำคำไม่รู้จัก และการรู้จำนิพจน์ระบุนาม (Named Entity Recognition) เพื่อให้สามารถสกัดคำตอบได้ถูกต้องมากขึ้น

5. กิตติกรรมประกาศ

ขอขอบคุณหน่วยภาษาศาสตร์คำนวณ (Thai Computational Linguistics Laboratory) ที่ให้ความอนุเคราะห์พจนานุกรมเชิงความหมายทีซีแอลสำหรับใช้ในงานวิจัยนี้ และคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือที่ให้ทุนอุดหนุนการวิจัยบางส่วน

เอกสารอ้างอิง

[1] Pasca, M. A. and Harabagiu, S. M. “High performance question/answering.” *Proceedings of the 24th international Conference on Research and Development in Information Retrieval (SIGIR-2001)*, pp. 366-374, New Orleans, Louisiana, USA, 2001.

[2] Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea,

R., Goodrum, R., G irji , R., and Rus, V. "LASSO: A tool for surfing the answer net." *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*, pp. 175-184, Maryland, USA, 1999.

[3] Lee, G.G., J. Seo, S. Lee, H. Jung, B-H. Cho, C. Lee, B-K. Kwak, J, Cha, D. Kim, J-H. An, H. Kim, and K. Kim. "SiteQ : Engineering High Performance QA System Using Lexico-Semantic Pattern Matching and Shallow NLP." *10th Text Retrieval Conference (TREC-10)*, pp. 437-446, Washington DC, USA, Nov. 2001.

[4] Dumais, S., Banko, M., Brill, E., Lin, J., and Ng, A. "Web Question Answering: Is More Always Better?." *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, pp. 291-298, Tampere, Finland, August 2002.

[5] M. M. Soubbotin and S. M. Soubbotin. "Patterns of Potential Answer Expressions as Clues to the Right Answer." *Proceedings of the 10thText Retrieval Conference (TREC10)*, pp. 175-182, Maryland, USA, 2001.

[6] Ittycheriah, A.; Franz, M.; Zu, W. and Ratnaparkhi, A. "IBM's Statistical Question Answering System." *9th Text Retrieval Conference(TREC 9)*, Maryland, USA, 2000.

[7] C. Monz. "From Document Retrieval to Question An swering." Ph.D Thesis, University of Amsterdam, 2003.

[8] เกศสุดา ตุริยากรณ์. "ระบบคำถาม-คำตอบ สำหรับข้อความภาษาไทย." วิทยานิพนธ์ปริญญาวิทยาศาสตร์มหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์, 2546.

[9] K. Sadakane, H. Imai. "Fast algorithms for k-word proximity search." *IEICE Trans.* Fundamentals E84-A (9) (September 2001) : 312-319.

[10] Mittrapiyanuruk P. and Sornlertlamvanich V. "The automatic Thai sentence extraction." *The Fourth Symposium on Natural Language Processing 2000*, pp. 23-28, Thailand, May 2000.

[11] Charoenpornsawat P. and Sornlertlamvanich V. "Automatic Sentence Break Disambiguation for Thai." *Proceedings of ICCPOL2001*, pp. 231-235, Korea, May 2001.

[12] Cortes C. and Vapnik V. "Support-vector networks." *Machine Learning*, 20 (November 1995) : 273–297.

[13] Chuleerat Jaruskulchai. "An Automatic Indexing for Thai Text Retrieval." Ph.D. Thesis, School of Engineering and Applied Science, George Washington University, 1998.

[14] Apache Lucene project. Available online at http://lucene.apache.org.

[15] Salton, G., Wong, A. and Yang, C. "A vector space model for automatic indexing." *Communications of the ACM*, 18 (November 1975) : 613–620.

[16] Salton, G. and Lesk, M. E. "Computer evaluation of indexing and text processing." *Journal of the ACM*, 15 (January 1968) : 8-36.

[17] Tiny SVM: Support Vector Machines. Available online at http://chasen.org/~taku/software/ TinySVM.

[18] YamCha: Yet Another Multipurpose Chunk. Available online at http://www.chasen.org/~taku/software/yamcha.

[19] V. Sornlertlamvanich, N. Takahashi and H. Isahara. "Building a Thai part-of-speech tagged corpus (ORCHID)". *Journal of the Acoustical of Society of Japan*, 20 (May 1999) : 189-198.

[20] ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ. SWATH : โปรแกรมตัดคำและกำกับหน้าที่คำภาษาไทยอัตโนมัติ. เว็บเพจ : http://www.links.nectec.or.th/download_thai.php.

[21] T. Charoenporn, C. Kruengkrai, V. Sornlertlamvanich and H. Isahara. "Acquiring Semantic Information in the TCL's Computational Lexicon." *Proceedings of the Fourth Workshop on Asia Language Resources*, IJCNLP-04, pp. 47-53, Sanya City, Hainan Island, China, 25 March 2004.

[22] อมรทิพย์ กวินปณิธาน. "การศึกษาบริบทบ่งบอกชื่อเฉพาะในภาษาไทยตามแนวภาษาศาสตร์คอมพิวเตอร์."

วิทยานิพนธ์ปริญญาอักษรศาสตร์มหาบัณฑิต บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2546.

[23] Voorhees, E., and Tice, D. M. 1999. "The TREC-8 Question Answering Track Evaluation." *Proceedings of The Eighth Text Retrieval Conference (TREC-8)*, Available online at http://trec.nist.gov/pubs/trec8/t8_proceedings.html.