วารสารพัฒนบริหารศาสตร์ ปีที่ 31 ฉบับที่ 4 ตุลาคม – ธันวาคม 2534

การประมวลผลภาษาไทย : การทำดรรชนีคำสำคัญ*

สัตตา ยินดีมาก โกรติ**

1. บทนำ

บทความนี้เป็นรายงานผลงานวิจัย การประมวลผลภาษาไทยตามลักษณะภาษาธรรมชาติ (natural language processing) เรื่องการทำดรรชนีคำสำคัญอัตโนมัติ (automatic keyword indexing) โดยการทดลองใช้เทคนิคที่ใช้กับภาษาอังกฤษผสมผสานกับเทคนิคการหาความถี่ของคำที่อยู่ติดกัน (neighborhood frequency) ซึ่งเป็นเทคนิคที่ได้พัฒนาขึ้นสำหรับงานวิจัย ทั้งนี้ มีวัตถุประสงค์เพื่อศึกษาความเป็นไปได้ของเทคนิคและวิธีการต่าง ๆ ในการที่จะช่วยวิเคราะห์และพัฒนาด้านการประมวลผลภาษาไทยตามลักษณะภาษาธรรมชาติ

การประมวลผลภาษาไทยด้วยคอมพิวเตอร์ ประสบปัญหา อุปสรรค นานัปการ เนื่องจากโครงสร้างภาษาไทยมีเอกลักษณ์ที่แตกต่างไปจากโครงสร้างภาษาอังกฤษทั้งในลักษณะการอ่านและเขียน กล่าวคือ ภาษาอังกฤษมีโครงสร้างอักขระเพียงระดับเดียว ในแต่ละตำแหน่งของอักขระจะมีอักขระเพียงตัวเดียวและอ่านจากซ้ายไปขวา อุปสรรคสำคัญอีกประการหนึ่งก็คือ ไม่มีช่องว่างระหว่างคำในภาษาไทย การแยกคำในภาษาไทยมิอาจกระทำได้โดยอาศัยหลักไวยากรณ์ (syntax) เท่านั้น แต่จะต้องอาศัยความหมายของคำ (semantics) ในประโยคนั้น ๆ ด้วย นอกจากนี้ คำในภาษาไทยยังมีลักษณะเป็นคำประสม (compound words) เป็นส่วนใหญ่อีกด้วย ด้วยเหตุผลดังกล่าวเหล่านี้ ทำให้การใช้คอมพิวเตอร์

---

*สรุปจากวิทยานิพนธ์ปริญญาเอกทาง Library and Information Science ณ Indiana University ผู้เขียนขอขอบคุณ รองศาสตราจารย์ ดร.วิชิต หล่อจีระชุณห์กุล ที่กรุณาเป็นอาจารย์ที่ปรึกษา

**อาจารย์ สำนักบรรณสารการพัฒนา สถาบันบัณฑิตพัฒนบริหารศาสตร์

ประมวลผลภาษาไทยตามลักษณะภาษาธรรมชาติมีน้อยมาก ในขณะเดียวกันก็ใช้วิธีการประมวลผลโดยการเขียนภาษาไทยด้วยอักขระโรมัน (transliteration) อย่างไรก็ตามวิธีนี้เป็นการฝืนธรรมชาติของคนไทยและต้องเป็นผู้รู้เท่านั้นจึงสามารถกระทำงานในลักษณะที่ฝืนธรรมชาติได้ อนึ่ง ประโยชน์โดยตรงก็มีน้อยมาก จากการวิจัยที่มหาวิทยาลัยโตเกียว (Shibayama, 1987) พบว่าวิธีเขียนภาษาไทยด้วยอักขระโรมันต้องใช้จำนวนครั้งในการบันทึกข้อมูล (key strokes) เพิ่มขึ้นถึง 42.9 เปอร์เซ็นต์ แต่จำนวนอักขระต่อนาทีของการบันทึกข้อมูลจะมากกว่าการบันทึกด้วยอักขระไทยโดยตรง 9.8 เปอร์เซ็นต์

ดรรชนี (index) เป็นเครื่องมือที่มีความสำคัญมากในการสืบค้นสารนิเทศ (information retrieval) ทั้งนี้ดรรชนีเป็นเครื่องชี้เรื่องที่ต้องการ การทำดรรชนีโดยทั่วไปแบ่งได้ 3 ประเภท คือ ดรรชนีผู้แต่ง (author index) ดรรชนีชื่อเรื่อง (title index) และดรรชนีหัวเรื่อง (subject index) (Borko and Bernier, 1978) ดรรชนีคำสำคัญ (keyword index) เป็นดรรชนีประเภทหนึ่งที่มีลักษณะคล้ายกับดรรชนีหัวเรื่อง แต่จะมีรายละเอียดมากกว่า เพราะครอบคลุมคำสำคัญทั้งหลายที่ปรากฏในเอกสารซึ่งสามารถให้จำนวนคำสำคัญได้มากกว่าและทันสมัยกว่า

เอกสาร (source document) ที่ใช้ทำดรรชนี อาจใช้ชื่อเรื่องเอกสาร (title) บทคัดย่อหรือสาระสังเขป (abstract) หรือตัวเอกสาร (full text) อย่างใดอย่างหนึ่ง หรือมากกว่าหนึ่งอย่าง การทำดรรชนีโดยระบบคอมพิวเตอร์เป็นเรื่องยากและซับซ้อน การใช้ตัวเอกสารทำดรรชนีสามารถให้ประสิทธิผลมากกว่าการใช้ชื่อเรื่อง หรือบทคัดย่อ แต่ทำให้เสียเวลาในการประมวลผลมากและผลประโยชน์ที่ได้รับอาจไม่คุ้มค่า สำหรับการใช้ชื่อเรื่องรวมกับบทคัดย่อก็สามารถให้ประสิทธิผลเท่ากับการใช้ตัวเอกสาร ขณะเดียวกันยังให้ประสิทธิภาพในการสืบค้นสารนิเทศตามลักษณะภาษาธรรมชาติของเอกสาร (free text search) อีกด้วย อย่างไรก็ตามการใช้บทคัดย่อของเอกสารเพียงอย่างเดียวในการทำดรรชนีก็สามารถให้ประสิทธิผลได้เพียงพอ (Van Rijsbergen, 1979)

การทำดรรชนีแบ่งได้ 2 ประเภท (Lancaster, 1979; Foskett, 1982) ประเภทแรกคือการดึงคำจากเอกสารโดยตรง (extraction method) ประเภทที่สอง คือ การกำหนดคำจากเนื้อหาเอกสาร (assignment method) ประเภทหลังต้องใช้ศัพท์ควบคุม (controlled vocabulary) สำหรับวิธีทำดรรชนีส่วนใหญ่จะใช้วิธีทางสถิติ (statistical) หรือวิธีทางภาษาศาสตร์ (linguistical) วิธีใดวิธีหนึ่ง (Boyce and Kraft, 1985)

การถือความถี่ของคำที่ปรากฏในเอกสาร (word-frequency based method) เป็นวิธีการทางสถิติวิธีหนึ่ง โดยมีหลักว่าคำใดที่ปรากฏบ่อยครั้ง หรือมีจำนวนความถี่สูงในเอกสาร คำนั้นจะมีความสำคัญต่อการชี้บ่งเนื้อหาสาระของเอกสาร ทั้งนี้จะต้องตัดคำที่ไม่สำคัญออกจากเอกสารเสียก่อน ซึ่งได้แก่คำที่ทำหน้าที่ เชื่อม ขยาย คำ วลี หรือประโยค (function word หรือ common word) ในภาษาอังกฤษจะมีอยู่ถึง 40 ถึง 50 เปอร์เซ็นต์ในตัวเอกสาร คำประเภทนี้มีอยู่ประมาณ 250 คำ เช่น a, an, the, for, in, on, at เป็นต้น (Salton and McGill, 1983) คำอื่น ๆ ที่ปรากฏน้อยครั้งในเอกสารเมื่อเปรียบเทียบจากจำนวนคำทั้งหมดก็ควรถูกตัดออกเช่นกัน (Damerau, 1965) Luhn (1957) เป็นบุคคลแรกที่เริ่มใช้ความถี่ของคำเป็นตัวกำหนดความสำคัญของคำ และได้มีการศึกษาวิจัยกันอย่างแพร่หลาย อาทิเช่น Baxendale (1958) Carroll and Roeloffs (1969) Pao (1978) Weinberg (1981) และอื่น ๆ อีกมาก Bookstein และ Swanson (1975) มีความเห็นว่าการถือความถี่ของคำในการทำดรรชนี เป็นวิธีการที่เหมาะสม สำหรับการใช้วิธีการทางภาษาศาสตร์ได้เริ่มในช่วงทศวรรษ 1970 (Sparck Jones and Kay, 1973) และต่อมาได้เน้นถึงการประมวลผลตามลักษณะภาษาธรรมชาติกันมาก อาทิเช่น Sparck Jones และ Wilks (1985) Doszkocs (1986)

คำที่มีความถี่สูงที่เลือกได้จากเอกสารจะต้องนำมาให้น้ำหนัก (weighting) อีกครั้งก่อนที่จะเลือกคำใดเป็นคำสำคัญ (keyword) มาตรการของการให้น้ำหนักอาจใช้การเปรียบเทียบความถี่ของคำในแต่ละเอกสาร (document frequency) หรือเปรียบเทียบกับเอกสารทั้งหมด (collection frequency) เป็นต้น โดยอาจพิจารณาความถี่จากรูปแบบใดรูปแบบหนึ่ง เช่น ใช้ความถี่สัมบูรณ์ (absolute frequency) ความถี่สัมพัทธ์ (relative frequency) ความถี่ผกกลับ (inverse document frequency) และอื่น ๆ เทคนิคของการให้น้ำหนักคำสำคัญมีมากหลาย (Sparck Jones, 1973, 1974 ; Salton and McGill, 1983) การใช้เทคนิคใดก็ตามขึ้นอยู่กับปัจจัยต่าง ๆ เป็นต้นว่าประเภทของเอกสาร ลักษณะการวิจัย สมมุติฐาน ฯลฯ Ro (1985) ได้ทดสอบอัลกอริทึมการให้น้ำหนักคำ 29 วิธี โดยใช้การสืบค้นสารนิเทศในการทดลอง และพบว่าไม่มีผลต่างอย่างมีนัยสำคัญระหว่างอัลกอริทึมทั้ง 29 วิธี

2. ขอบเขตและข้อจำกัด

ดังที่ได้กล่าวมาแล้วว่าอักขระไทยเขียนติดต่อกันโดยไม่มีช่องว่างระหว่างคำ (inter-word space) และยังไม่มีวิธีการที่จะสามารถกำหนดคำไทยที่สมบูรณ์ เพื่อใช้ในการประมวลผล

ด้วยคอมพิวเตอร์ได้ การกำหนดคำในงานวิจัยนี้จึงใช้วิธีการกำหนดออกเป็น token ซึ่งวิชิต หล่อจีระชุณห์กุล และเจริญ คุวินทร์พันธุ์ ได้พัฒนาอัลกอริทึมขึ้นเมื่อปี พ.ศ. 2524 "token" ในที่นี้หมายถึงอักขระไทยที่เขียนติดกันซึ่งอ่านได้อาจมีความหมายหรือไม่มีความหมายก็ได้ token หนึ่ง ๆ อาจประกอบด้วย พยางค์เดียวหรือหลายพยางค์ก็ได้ (syllables)

เอกสารหรือข้อมูล (source document) ที่ใช้ทำดรรชนีเป็นบทคัดย่อวิทยานิพนธ์ปริญญาโทสาขาธุรกิจ จำนวน 100 บท เนื่องจากเป็นการทดลองวิจัยครั้งแรกจึงเห็นสมควรว่า ควรทดสอบกับเอกสารสาขาวิชาเดียวกัน ความยาวเฉลี่ยของบทคัดย่อประมาณหนึ่งหน้าครึ่งซึ่งคาดว่าเพียงพอต่อการแสดงเนื้อหาสาระสำคัญ

การทำดรรชนีคำสำคัญในที่นี้ใช้วิธีดึงคำจากเอกสารโดยตรง ไม่ต้องอาศัยสติปัญญาของมนุษย์ในการวิเคราะห์เนื้อหาเอกสาร ไม่ต้องใช้คู่มือหัวเรื่องบัญชีศัพท์สัมพันธ์ (thesaurus) หรือศัพท์ควบคุม (controlled vocabulary) แต่จะใช้คำในเอกสารโดยตรงและต้องเป็นคำที่มีความถี่สูงซึ่งคำประเภทนี้ถือว่าสามารถสื่อความหมายของเนื้อเรื่องได้ ทั้งนี้จะยกเว้นคำสรรพนาม วิเศษณ์ บุพบท สันธาน ฯลฯ การสร้างกลุ่มคำ หรือวลี ทำโดยการนำคำที่อยู่ติดกันมารวมเข้าเป็นข้อความเดียวกัน โดยพิจารณาดูจากจำนวนครั้งที่คำอยู่ติดกัน หรือเรียกว่า ความถี่ของ neighborhood (neighborhood frequency) เทคนิคนี้ได้พัฒนาขึ้นใช้สำหรับการสร้างกลุ่มคำ และปรากฏว่าเป็นเทคนิคที่มีประสิทธิภาพมาก การให้น้ำหนักกลุ่มคำที่สร้างขึ้นเพื่อคัดเลือกคำที่มีน้ำหนักมากเป็นคำสำคัญนั้น จะเน้นน้ำหนักของคำประสมเนื่องจากคำไทยส่วนใหญ่เป็นคำประสม

3. เทคนิคและวิธีการ

การหาความถี่ของคำในเอกสาร ขั้นแรกต้องทำการแยกอักขระในเอกสารออกเป็นคำ ๆ ซึ่งวิธีแยกคำในภาษาไทยยังมิอาจกระทำได้โดยสมบูรณ์จึงใช้วิธีแยกออกเป็น token แทน ขั้นที่สองคือ การตัดคำที่ไม่สำคัญ ในที่นี้ คือ การตัด token ที่เป็นคำยกเว้น ขั้นที่สามคือการคำนวณความถี่ของ token ขั้นที่สี่ คือ การนำ token ที่มีความถี่สูงจากขั้นที่สามมากำหนดหา token ตัวที่อยู่ติดกันเพื่อนำมาประกอบเข้าเป็นคำ หรือวลี ขั้นที่ห้า คือ การนำคำหรือวลีที่เลือกได้มาให้น้ำหนักอีกครั้งเพื่อเลือกคำที่มีน้ำหนักมากที่สุดเป็นดรรชนีคำสำคัญ และขั้นสุดท้ายคือ การกำหนดจำนวนคำสำคัญต่อเอกสารซึ่งในทางปฏิบัติจะขึ้นอยู่กับความเหมาะสม และลักษณะงานของแต่ละงาน

3.1 การแยกอักขระไทยเป็น Token

อัลกอริทึมสำหรับแยกอักขระไทยเป็น token (Thai-Syllable Separation Algorithm) จะไม่ขอกล่าวในที่นี้ บทคัดย่อที่ใช้ประมวลผลจะรวบรวมอยู่ในแฟ้มข้อมูล TF_1 (input text) และประมวลผ่านอัลกอริทึมเพื่อแยกเป็น token ถ้ามีช่องว่างระหว่างคำในแฟ้มข้อมูลเข้า จะใช้สัญลักษณ์ "*" แทน คือ เป็นเทคนิคอย่างหนึ่งเพื่อชี้ว่ามีการสิ้นสุดของข้อความในส่วนนั้น ๆ เพื่อแสดงว่ามีการจบของคำ วลี หรือ ประโยคตามที่ปรากฏอยู่ในเอกสาร ผลที่ประมวลได้จากแฟ้มข้อมูล TF_1 จะจัดเก็บในแฟ้มข้อมูล TF_2

3.2 การตัด Token ที่ไม่สำคัญ

การตัดคำที่ไม่สำคัญหรือคำยกเว้น (stopword deletion) เป็นเทคนิคอย่างหนึ่งที่นำมาใช้ โดยทั่วไปแล้วข้อความต่าง ๆ จะประกอบด้วยคำที่สื่อความหมายได้ตรงกับเนื้อหาเอกสาร (content bearing word/significant word) และคำที่ไม่สามารถสื่อความหมายได้ตรง แต่จะทำหน้าที่ เชื่อม ขยาย ประกอบคำอื่น ๆ วลี หรือ ประโยคที่เรียกว่า (function word/ common word) เหตุผลของการตัดคำเหล่านี้ก็เพื่อลดขนาดคำ (word size) ให้สั้นและกระทัดรัด คำไม่สำคัญจะรวบรวมไว้ในแฟ้มข้อมูล Stopword File (SF_1) คำเหล่านี้ผู้วิจัยได้รวบรวมขึ้นเองโดยอาศัยหลักการของคำไม่สำคัญในภาษาอังกฤษ ซึ่งประกอบด้วย

3.2.1 คำสรรพนาม คุณศัพท์ วิเศษณ์ บุพบท สันธาน เช่น ก็ดี ก็ได้ ก็ตาม ทั้งนี้ ทั้งนั้น ทั้งหลาย

3.2.2 อักขระโรมัน (Roman letters) A - Z

3.2.3 ตัวเลขทั้งไทยและอารบิค 0 – 9

3.2.4 เครื่องหมายและสัญลักษณ์ต่าง ๆ เช่น [, (, ;, :, ?, X, ฯลฯ

การรวบรวมคำเพื่อสร้างเป็นแฟ้มข้อมูลคำไม่สำคัญ (Stopword File SF_1) มีปัญหาในการเลือกคำมากเพราะคำที่ได้ใจความหรือคำประสมนั้นจะมีทั้งคำที่สามารถให้ความหมายและคำที่ไม่ให้ความหมายต่อเนื้อหาสาระของเอกสาร เช่น คำว่า "ก็ดี" เมื่อใช้เป็นคำไม่สำคัญก็อาจมีผลกระทบต่อคำอื่นได้ เป็นต้นว่า มีผลให้คำว่า "ดี" ในคำ "ดีบุก" ถูกตัดออก ทำให้คำที่เหลืออยู่เป็นคำไม่สมบูรณ์ในส่วนของข้อความนั้นแต่หากคง "ก็ดี" ไว้ คำนี้อาจถูกเลือกเป็นคำสำคัญได้ เพราะคำประเภทนี้มักจะปรากฏอยู่บ่อยครั้งคือจะมีความถี่สูง

Stopword (SF_1) จะถูกประมวลแยกออกเป็น token และจัดไว้ในแฟ้มข้อมูล SF_2 โดยวิธีการเดียวกับการประมวลข้อความใน TF_1 ให้เป็น token ใน TF_2 แฟ้มข้อมูลทั้งสองคือ SF_2 และ TF_2 จะนำประมวลเข้าด้วยกัน token ใด ใน TF_2 ที่เหมือนกับ token ใน SF_2 จะถูกตัดออก และแทนด้วยสัญลักษณ์ "*" เพื่อเป็นการชี้บอกขอบเขตของคำ อันประกอบด้วยหลาย ๆ token ที่อยู่ระหว่างสัญลักษณ์ "*" หรืออีกนัยหนึ่งแสดงขอบเขตของข้อความที่มีความหมาย แฟ้มข้อมูลที่ตัด token ออกแล้วจะจัดเก็บไว้ในชื่อ TF_3 เพื่อนำมาแจงความถี่ของ token แต่ละตัวต่อไป (ดูตัวอย่างหน้า 15)

3.3 การเลือก Token ที่มีความถี่สูง

Token ในแฟ้มข้อมูล TF_3 จะนำมาแจงความถี่เพื่อทำการเลือกเฉพาะ token ที่มีความถี่สูง (relatively-high frequency token) ส่วน token ที่มีความถี่ต่ำจะถูกตัดทิ้ง การเลือก token จะใช้เปอร์เซ็นต์ความถี่สะสม (cumulative frequency percentage) เป็นเกณฑ์ โดยมีขั้นตอนดังนี้

3.3.1 การแจงความถี่ (Frequency of Occurrence)

3.3.1.1 Token แต่ละตัวเมื่อแจงความถี่แล้วจะถูกจัดเรียงตามลำดับความถี่สูงไปหาต่ำ ถ้ามีความถี่เท่ากันจะเรียงตามพยัญชนะ ก – ฮ

3.3.1.2 Token ที่มีความถี่เท่ากับ 1 จะถูกตัดทิ้ง โดยถือว่าความถี่เท่ากับ 1 ไม่มีนัยสำคัญต่อการชี้บ่งเนื้อหาเอกสาร

3.3.2 การใช้เปอร์เซ็นต์ความถี่สะสม (Cumulative Frequency Percentage – CFP)

เพื่อที่จะคัดเลือกจำนวน token ให้ได้สัดส่วนที่เหมาะสมกับความถี่ทั้งหมด ดังนั้น จะเลือก token ที่มีความถี่สะสมอยู่ระหว่าง 70 เปอร์เซ็นต์ จากความถี่สะสม 100 เปอร์เซ็นต์ของ token ที่เหลืออยู่

3.3.2.1 เลือก token ที่มีความถี่สะสม ต่ำกว่าหรือเท่ากับ 70 เปอร์เซ็นต์ (threshold = 0.7) หรือ CFP < = 0.7 โดยถือว่า token ที่อยู่ในเกณฑ์นี้มีความถี่สูงและมีนัยสำคัญเพียงพอ (significant token) ค่า threshold นี้เป็นค่าที่ได้จากการทดลองหนึ่งในสามของเอกสารทั้งหมด

3.3.2.2 Token ที่มีความถี่สะสมเกินกว่า 70 เปอร์เซ็นต์ แต่ถ้ามีความถี่

เฉพาะตัว (frequency of individaul token) เท่ากับ token ที่อยู่ในเกณฑ์ 70 เปอร์เซ็นต์ ก็จะถูกเลือกไว้เช่นกัน ด้วยเหตุผลที่ว่า token ใดที่มีความถี่เฉพาะตัวเท่ากันย่อมมีนัยสำคัญเท่ากัน ซึ่งจะทำให้มีจำนวน token ที่ถูกเลือกเพิ่มขึ้นและความถี่สะสมก็จะสูงขึ้นตามลำดับ

ตัวอย่าง จำนวน token ที่มีความถี่สูงจากเอกสารที่มีขนาดยาวและสั้น

จำนวน token ที่เลือกได้ต่อเอกสาร	ความถี่สูงสุด	ความถี่ต่ำสุด
เอกสารยาว (3 เรื่อง)		
33	87	6
42	60	5
36	51	5
เอกสารสั้น (3 เรื่อง)		
29	9	2
33	9	2
43	10	2

ตัวอย่างข้างบน แสดงถึงจำนวน token ซึ่งเลือกมาจากเอกสารที่มีขนาดความยาวมากสุดและต่ำสุด จาก 100 เอกสาร จะเห็นได้ว่า ความถี่สูงสุดและต่ำสุดของ token นั้น มีความแตกต่างกันมาก แต่จำนวนของ token ที่ถูกเลือกจากแต่ละเอกสารจะมีขนาดใกล้เคียงกัน token เหล่านี้จะนำมาเก็บใน matrix เพื่อหาความถี่ของ token ที่อยู่ติดกันสำหรับการสร้างคำและกลุ่มคำ

ตัวอย่าง ขั้นตอนการเลือก token ที่มีความถี่สูง

	บทคัดย่อ 1	บทคัดย่อ 2
จำนวน token ทั้งหมด	691	717
จำนวน token ที่ถูกกำจัด (stopword)	360 (52%)	335 (46%)
จำนวน token ที่ถูกกำจัด (freq. 1)	163	159
จำนวน token ที่เลือกได้ (CFP < = 0.7)	22	25
จำนวน token ที่เลือกเพิ่ม (CFP > 0.7)	9	7
CFP (เปอร์เซ็นต์ความถี่สะสมที่ใช้จริง)	0.85	0.83

3.4 การสร้างคำและกลุ่มคำโดยการใช้ความถี่ของ Neighborhood (Neighborhood Frequency)

เทคนิคในการหาความถี่ของ token ที่อยู่ติดกันได้พัฒนาขึ้นและถือได้ว่าเป็นกลไกอันสำคัญในการสร้างคำและกลุ่มคำ กล่าวคือจะตรวจสอบว่ามี token ใดที่อยู่ติดกันบ้าง อยู่ติดกันกี่ครั้ง จำนวนครั้งที่อยู่ติดกัน จะเรียกว่าความถี่ของ neighborhood ถ้ามีความถี่อยู่ในเกณฑ์ที่กำหนดไว้ token เหล่านี้จะนำมาเรียงกันเป็นคำ (concatenation of tokens) วิธีการมีดังนี้

3.4.1 การสร้าง Matrix เพื่อคำนวณความถี่ของ Neighborhood

Matrix 2 มิติ (2 dimensional array) เป็น matrix ที่สร้างขึ้นเพื่อจัดเก็บ token ที่เลือกมาจากความถี่สะสมต่ำกว่าหรือเท่ากับ 0.7 โดยจัดเก็บไว้ทั้งด้านแถว (rows) และด้านสดมภ์ (columns) ตัวอย่างเช่นในเอกสารฉบับหนึ่งมีจำนวน token ที่เลือกได้ 24 ตัว token ทั้ง 24 ตัวนี้จะปรากฏอยู่ในแถวที่ 1 ถึง 24 และในสดมภ์ที่ 1 ถึง 24 เช่นกัน token ที่อยู่ในแถวจะทำหน้าที่เป็น token นำ (starting token) และ token ในสดมภ์จะทำหน้าที่เป็น token ตาม (adjacent token)

แฟ้มข้อมูล TF_3 จะนำมาอ่านเพื่อตรวจหาความถี่ของ token นำ และ token ตาม วิธีการหาความถี่ของ neighborhood คือจะอ่าน token ใน TF_3 ว่าเป็น token ที่มีอยู่ในแถวหรือไม่ ถ้าใช่ token นั้นจะเป็น token นำ และ token ใน TF_3 ตัวต่อไปจะถูกตรวจว่าเป็น token ที่อยู่ในสดมภ์หรืออีกนัยหนึ่งคือ token ตาม หรือไม่ ถ้าไม่ใช่ก็จะทำหน้าที่เป็น token นำ และอ่าน token ใน TF_3 ต่อไปอีก เพื่อตรวจหาการเป็น token ตาม จนพบเครื่องหมาย "*" ที่แสดงการสิ้นสุดของหน่วยข้อความที่อยู่ติดกัน เป็นดังนี้เรื่อยไปจนจบแฟ้มข้อมูล TF_3 จำนวนครั้งที่ token นำ และ token ตาม อยู่ติดกันคือความถี่ของ neighborhood ($F_{i,j}$) ค่าของความถี่จะเก็บไว้ใน matrix เช่น แถวที่ 2 สดมภ์ที่ 3 มีค่า = 5, นั่นคือ ($F_{2,3}$) = 5 หมายความว่า $token_2$ และ $token_3$ อยู่ติดกัน 5 ครั้ง

3.4.2 การสร้างคำที่ประกอบด้วย Token สองตัว

ค่าความถี่ของ neighborhood ($F_{i,j}$) ที่ได้จากการคำนวณใน matrix จะใช้เป็นเกณฑ์ในการสร้างคำของ token นำ และ token ตาม โดยมีวิธีการดังนี้

3.4.2.1 $F_{(i,j)} = 1$ จะถูกตัดทิ้ง โดยคำนึงว่าการอยู่ติดกันของ token นำ และ token ตาม เพียงครั้งเดียวยังสำคัญไม่เพียงพอที่สร้างเป็นคำ

3.4.2.2 คำที่ถูกสร้างจะต้องมีค่า normalized neighborhood frequency ($N_{(i,j)}$) มากกว่าหรือเท่ากับ 0.3 กล่าวคือ

$$N_{(i,j)} = F_{(i,j)}/F_i = 0.3$$

F_i = ความถี่เฉพาะของ $token_i$(token นำ)

การกำหนดเกณฑ์ (threshold) >= 0.3 ได้มาจากการทดลองส่วนหนึ่งของเอกสาร เพื่อให้สามารถได้คำที่มีความถี่ของการอยู่ติดกันหรือการเป็น neighborhood ของ token นำ และ token ตาม ที่มากครั้งพอสมควร สมมุติว่า $F_{(i,j)}$ มีค่าเท่ากับ 5 และ F_i มีค่าเท่ากับ 15

$$N_{(2,3)} = F_{(2,3)}/F_2 = 5/15 >= 0.3$$

ดังนั้น $token_2$ และ $token_3$ จะถูกสร้างให้เป็นคำ

3.4.2.3 ความถี่ของคำที่สร้างขึ้นในกรณีดังกล่าว เรียกว่า $NHF_{(i,j)}$ และจะมีค่าเท่ากับค่า $F_{(i,j)}$ เช่น $F_{(2,3)} = 5$ หมายถึง คำที่เป็น $token_2 + token_3$ มีความถี่ของคำ $NHF_{(2,3)} = 5$

3.4.3 การสร้างคำที่ประกอบด้วย Token หลายตัว

การสร้างคำที่ประกอบด้วย token หลายตัว คือ การนำ token คู่ (token นำ + token ตาม) ที่มี $NHF_{(i,j)} >= 0.3$ มาเรียงเข้าด้วยกัน (concatenation of token-pairs) จะเห็นได้ว่า token ตัวใดตัวหนึ่งอาจประกอบอยู่ในหลาย ๆ token คู่ ซึ่งอาจจะทำหน้าที่เป็น token นำ หรือ token ตาม เช่น $token_i$ อาจอยู่ใน ($token_i + token_j$) และใน ($token_h + token_i$) วิธีเรียง token เข้าด้วยกันจะพิจารณา token ที่ทำหน้าที่เป็นทั้ง token ตาม และ token นำ เมื่อเรียงแล้วจะได้คำใหม่ คือ ($token_h + token_i + token_j$) เป็นดังนี้เรื่อยไป

ความถี่ของคำ NHF_{Phrase} จะใช้ค่าความถี่ต่ำสุดของคำ เพราะถือว่าเป็นค่าสูงสุดที่เป็นไปได้ของคำที่จะมีได้

$$NHF_{Phrase} = \text{minimum}\ [\ NHF_{(i,j)}, NHF_{(j,k)}, \ldots NHF_{(m,n)}\ ]$$

ตัวอย่าง เช่น

(1) $(t_2 + t_3)$, $NHF_{(2,3)} = 15$

(2) $(t_5 + t_2)$, $NHF_{(5,2)} = 14$

(3) $(t_{13} + t_5)$, $NHF_{(13,5)} = 5$

เมื่อนำ token แต่ละคู่มาเรียงกัน จะมีกลุ่มคำใหม่ ดังนี้ คือ

(4) $(t_5 + t_2 + t_3)$, $NHF_p = \text{min. } [\, NHF_{(5,2)}, NHF_{(2,3)} \,] = 14$

(5) $(t_{13} + t_5 + t_2)$, $NHF_p = \text{min. } [\, NHF_{(13,5)}, NHF_{(5,2)} \,] = 5$

(6) $(t_{13} + t_5 + t_2 + t_3)$, $NHF_p = \text{min. } [\, 5, 14, 15 \,] = 5$

3.4.4 การกำจัดคำและส่วนคำซ้ำซ้อน

คำ หรือ กลุ่มคำ (pairs, phrases) ที่สร้างขึ้นในขั้นนี้ยังมีความซ้ำซ้อนอยู่ เพราะ token หนึ่ง ๆ อาจประกอบอยู่ในหลายคำ หรือกลุ่มคำ (ขอใช้คำว่า term แทน) จึงต้องกำจัดส่วนของคำที่ซ้ำซ้อน โดยวิธีการดังนี้

3.4.4.1 Term ใดที่มี token นำ เหมือนกัน ให้คงไว้เฉพาะ term ที่ใหญ่กว่า

3.4.4.2 Term ที่มี token นำ ต่างกันแต่มีส่วนของ term เหมือนกัน ให้คงไว้ทั้ง term ใหญ่ และ term เล็ก **ถ้า** term เล็ก มีค่าความถี่ของคำ (NHF) สูงกว่า NHF ของ term ใหญ่ มิฉะนั้นให้คงไว้เฉพาะ term ใหญ่

ตัวอย่าง เช่น

(1) $(t_2 + t_3)$, $NHF_{(i,j)} = 15$

(2) $(t_5 + t_2)$, $NHF_{(i,j)} = 14$ *

(3) $(t_{13} + t_5)$, $NHF_{(i,j)} = 5$ *

(4) $(t_5 + t_2 + t_3)$, $NHF_{(i,j)} = 14$

(5) $(t_{13} + t_5 + t_2)$, $NHF_p = 5$ *

(6) $(t_{13} + t_5 + t_2 + t_3)$, $NHF_p = 5$

Token นำ เหมือนกัน

Term (3), (5) และ (6) มี token นำ (t_{13}) เหมือนกัน และมี NHF_5 เท่ากันทั้งสาม term ซึ่งแสดงว่า term (3) และ term (5) เป็นส่วนประกอบ (component) ของ term (6) จึงต้องตัด term (3) และ (5) ซึ่งเล็กกว่าออก และสำหรับ term (2) ก็จะถูกตัดทิ้งเช่นกัน เพราะเป็นส่วนประกอบของ term (4)

Token นำ ต่างกัน

Term (1) เป็นส่วนประกอบ term (4) และ (6) แต่ term (1) มี NHF สูงกว่า term (4) และ (6) เช่นเดียวกับ term (4) เป็นส่วนประกอบของ term (6) แต่มี NHF สูงกว่า ในกรณีนี้ term (1) และ term (4) จึงยังคงไว้ จากตัวอย่าง term ที่ยังคงอยู่คือ (1), (4) และ (6)

3.4.5 การกำจัดความถี่ซ้ำซ้อน

ความถี่ของ token และ term ยังคงมีความซ้ำซ้อนอยู่ จึงต้องทำการกำจัดความถี่ที่ซ้ำซ้อนออกเพื่อจะได้ความถี่ที่เป็นจริง (stand alone frequency) ของแต่ละ token และ term โดยการลบความถี่ออกทุกครั้งที่ token ใด หรือ term ใด เป็นส่วนประกอบของ term อื่น วิธีนี้จะเริ่มจาก term ที่ใหญ่กว่าไปยัง term ที่เล็กกว่า เช่น พิจารณาจาก term ที่คงไว้ในหัวข้อ 3.4.4

term	(1)	$(t_2 + t_3)$,	$NHF_{(i,j)}$	=	15
	(4)	$(t_5 + t_2 + t_3)$,	NHF_p	+	14
	(6)	$(t_{13} + t_5 + t_2 + t_3)$,	NHF_p	=	5
token		t_2 มี total frequency		=	15
		t_3 ,,		=	15
		t_5 ,,		=	14
		t_{13} ,,		=	6

จะเห็นได้ว่า term (4) เป็นส่วนประกอบของ term (6) อยู่ 5 ครั้ง เพราะฉะนั้นความถี่ (NHF) ของ term (4) จริง ๆ จะมีเพียง 9 ครั้ง คือ 14 5 = 9 และ term (1) เป็นส่วนประกอบของ term (6) อยู่ 5 ครั้งเช่นกัน และ term (4) อีก 9 ครั้ง ดังนั้น $NHF_{(i,j)}$ ของ term (1) ที่จะมีได้โดยตัวเอง คือ 15 – 5 – 9 = 1 ดังนี้เป็นต้น

วิธีการ	$(t_{13} + t_5 + t_2 + t_3)$	NHF_p	=	5	final freq.	=	5
	$(t_5 + t_2 + t_3)$	NHF_p	–	14	14 – 5	=	9
	$(t_2 + t_3)$	$NHF_{(i,j)}$	=	15	15 5 – 9	=	1
	t_2	total freq.	–	15	15 – 9 – 5 – 1	=	0
	t_3	,,	=	15	15 9 – 5 – 1	=	0

t_5	,,	=	14	14 · 5 − 9	=	0
t_{13}	,,	=	6	6 − 5	−	1

3.4.6 การให้น้ำหนักคำ (Term Weighting)

การจัดน้ำหนักให้กับคำเป็นหลักการสำคัญประการหนึ่งเพื่อที่จะจัดเรียงลำดับความสำคัญของ term เพื่อเลือกเฉพาะ term ที่มีน้ำหนักมาก หรือความถี่สูงไว้เป็นสำคัญ (keyword) ของเอกสาร ในที่นี้ใช้ 2 วิธีรวมกัน คือ

3.4.6.1 พิจารณาจากความถี่สัมบูรณ์ของคำ (absolute frequency) โดยคำนึงว่าผู้เขียนต้องการเน้นเนื้อหาเกี่ยวกับเรื่องอะไรก็จะเขียนคำ ๆ นั้นบ่อยครั้ง ในกรณีนี้น้ำหนักของ term คือความถี่ของ term นั้น

3.4.6.2 พิจารณาจากจำนวน token ที่ประกอบเข้าเป็น term เนื่องจากคำไทยส่วนใหญ่เป็นคำประสมซึ่งอาจประกอบด้วยหลาย token และอาจทำให้มีความถี่ของ term ต่ำได้ การใช้ความถี่สัมบูรณ์เพียงอย่างเดียวอาจไม่เพียงพอต่อการให้น้ำหนักคำได้ เพราะ term ที่มีเพียง token เดียว ถ้ามีความถี่มากก็จะมีน้ำหนักมาก แต่อาจได้ใจความไม่สมบูรณ์

จากข้อพิจารณาสองประการนี้ การให้น้ำหนักคำ จึงใช้ความถี่ของ term คูณกับจำนวน token ที่ประกอบเป็น term นั้น

$$\text{weight}_{ij} = \text{frequency}_{ij} * (\text{number of token})_{ij}$$

ตัวอย่าง จากหัวข้อ 3.4.5

	ความถี่	**จำนวน token**	**น้ำหนัก**
$(t_{13} + t_5 + t_2 + t_3)$	5	4	20
$(t_5 + t_2 + t_3)$	9	3	27
$(t_2 + t_3)$	1	2	1
t_{13}	1	1	1

3.4.7 การกำหนดจำนวนคำสำคัญ (Cutoff Point)

การกำหนดจำนวนคำสำคัญต่อเอกสารโดยทั่วไปจะใช้วิธีการ กำหนดจำนวนคำเอง ตามความต้องการและความเหมาะสมของแต่ละงาน โดยเลือก term ตามน้ำหนักมากลงมาตามลำดับ

4. บทวิเคราะห์

4.1 การประเมินผลคำสำคัญ

โดยทั่วไปการประเมินผลคำสำคัญอัตโนมัติจะกระทำโดยใช้การทดลองสืบค้นสารนิเทศ (information retrieval experiment) และพิจารณาจากสัดส่วนของ recall และ precision (Sparck Jones, 1981) recall คือ จำนวนคำที่เกี่ยวข้องทั้งหมดที่สืบค้นได้ (relevant items) ส่วน precision คือจำนวนคำที่เกี่ยวข้องจริง (only relevant items) ระบบการทำดรรชนีที่มีประสิทธิผลมาก (optimal system) สัดส่วนของ recall และ precision ควรใกล้เคียง 0.80 ปัจจัยที่มีผลต่อ recall และ precision คือ หนึ่ง สามารถครอบคลุมคำสำคัญได้มาก (exhausitivity) ซึ่งจะทำให้ได้ระดับ recall สูง แต่ก็จะทำให้โอกาสที่มีคำที่ไม่เกี่ยวข้อง (non-relevant) สูงขึ้นด้วย ข้อสอง คือ สามารถให้คำที่เจาะจงชัดเจน (specificity) ซึ่งจะทำให้ได้ระดับ precision สูง แต่ก็จะทำให้ระดับของ recall ต่ำลงได้

เนื่องจากยังไม่ได้มีการทดลองสืบค้นสารนิเทศจากคำสำคัญอัตโนมัติที่ได้กล่าวมา การประเมินผลจึงใช้การเปรียบเทียบกับคำสำคัญที่ทำด้วยคน (manual keyword) และใช้สัดส่วนของ recall และ precision เป็นตัวชี้ในกรณีนี้ คำสำคัญที่ทำด้วยคนจึงมีค่าของ recall และ precision เต็มคือ 1

เมื่อกำหนดจำนวนคำสำคัญอัตโนมัติเอกสารละ 10 คำจาก 100 เอกสาร จะได้จำนวนคำที่ใช้ประเมินผล 1,000 คำ

	เกี่ยวข้อง	**ไม่เกี่ยวข้อง**	**รวมคำ**
มี	(a)	(b)	(a + b)
	537	463	1,000
ไม่มี	(c)		
	322	–	–
รวมคำ	(a + c)		
	859	–	–

ค่าของ recall = a/(a + c) = 537/(537 + 322) = .625

ค่าของ precision = a/(a + b) = 537/(537 + 463) = .537

อธิบายได้ว่าระบบการทำดรรชนีอัตโนมัติสามารถสร้างคำที่เกี่ยวข้องได้ .625 และมีคำที่เกี่ยวข้องจริง .537 นับว่าได้ผลเป็นที่น่าพอใจยิ่ง ในต่างประเทศใช้การสืบค้นสารนิเทศในการเปรียบเทียบระบบคำสำคัญอัตโนมัติ กับระบบคำสำคัญที่ทำด้วยคนพบว่าสัดส่วนของ recall และ precision ของทั้งสองระบบอยู่ระหว่าง .55 ถึง .65 (Salton, 1969)

ระดับของ precision และ recall ในงานวิจัยนี้สามารถปรับปรุงให้สูงขึ้นได้โดยการสร้างแฟ้มข้อมูลคำพิเศษเพียง 20 คำ เพื่อใช้ตัดคำหลังจากให้น้ำหนักคำเสร็จแล้ว จากคำสำคัญอัตโนมัติ 1,000 คำ พบว่ามีคำที่ไม่สมบูรณ์ หรือไม่มีความหมาย หรือเป็นคำประเภท function word พิจารณาเฉพาะ 20 คำที่มีความถี่สูง ได้ความถี่รวมถึง 186 หรือ 18.6 เปอร์เซ็นต์ คำพวกนี้ได้แก่ ประ, ผู้, สามารถ, สำ, จำ ฯลฯ ซึ่งส่วนมากประกอบคำอื่น ๆ เช่น "ผู้" ใน "ผู้บริโภค "สำ" ใน "มันสำปะหลัง" เป็นต้น จึงต้องคงไว้ในตอนแรก หากตัดคำเหล่านี้ออก ผลคือจำนวนคำที่ไม่เกี่ยวข้อง 463 คำจะลดลงเหลือ 277 คำ และในขณะเดียวกันก็จะได้จำนวนคำที่เกี่ยวข้องเพิ่มขึ้นด้วย

4.2 เอกสารที่ใช้ทำดรรชนี

ประเด็นสำคัญอีกประการหนึ่ง พบว่า เอกสารที่ใช้ในการทำดรรชนีคำสำคัญ คือ บทคัดย่อ สามารถให้คำสำคัญได้เพียงพอและตรงกับเนื้อหาสาระไม่จำเป็นต้องใช้ตัวเนื้อหาทั้งหมดของเอกสาร ข้อที่ควรปรับปรุง คือ ลักษณะการเขียนบทคัดย่อควรใช้คำที่กระทัดรัดชัดเจน ทั้งนี้เพื่อประโยชน์ของการประมวลผลภาษาไทยรวมถึงการสืบค้นสารนิเทศที่มีประสิทธิภาพในอนาคต

4.3 Token (Token Identifiers)

การใช้คอมพิวเตอร์แบ่งคำภาษาไทยที่ได้ใจความสมบูรณ์เป็นเรื่องที่กระทำได้ยากมาก อัลกอริทึมที่ใช้เป็น token ก็ยังมีข้อบกพร่องอยู่เพราะยังไม่สามารถแบ่งคำไทยที่เหมือนกันออกเป็น token ได้เหมือนกันทุกประการ (identical token) ทุกครั้ง กล่าวคือ อาจมีอักขระนำหน้าหรืออักขระตามหลังติดมากับ token ซึ่งทำให้ token ที่ควรจะเหมือนกันต้องต่างกันออกไป อย่างไรก็ตาม ผลลัพธ์ที่ได้มีทั้งส่วนดีและส่วนเสีย การปรับปรุงอัลกอริทึมเพื่อแยก token ให้ดีขึ้นก็สามารถกระทำได้ แต่จะทำให้เสียเวลาคอมพิวเตอร์มาก และต้องใช้เวลาในการศึกษาวิจัยมาก

ส่วนดี

คำบางคำที่ใช้เป็น stopword แต่ในบางครั้งเมื่อรวมกับคำอื่นแล้วสามารถให้ความหมายได้สมบูรณ์ชัดเจนขึ้น ตัวอย่างเช่น คำว่า "การ" เมื่อติดมากับ token ก็จะช่วยขยายคำทำให้ได้คำที่มีใจความสมบูรณ์ เช่น **การ**ทำงาน, ผล**การ**ปฏิบัติงาน, สวัสดิ**การ**, สถานีบริ**การ**น้ำมัน ถ้าหาก "การ" ถูกแยกเป็น token เดี่ยวก็จะถูกตัดออกเพราะ "การ" เป็น stopword

ส่วนไม่ดี

เมื่อคำถูกแบ่งออกเป็น token ที่ไม่เหมือนกัน จะทำให้จำนวนความถี่ของ token ต่ำลงกว่าที่ควรจะเป็น และมีผลกระทบต่อการนำ token มาสร้างเป็นคำด้วย เช่น token "แชมพู" และ "ผม" ปรากฏว่ามี token "แชมพูของ" และ "ผมมี" ปรากฏอยู่ด้วย ทำให้กลายเป็น token คนละตัว ความถี่ของ token "แชมพู" และ "ผม" ก็จะเปลี่ยนไป

4.4 ความถี่สะสม

การใช้เกณฑ์ความถี่สะสม ทำให้สามารถเลือกจำนวน token จากแต่ละเอกสารได้อย่างเหมาะสม ไม่ว่าเอกสารนั้นจะสั้นหรือยาว และยังรวมถึง token ที่มีความถี่เฉพาะตัวเท่ากัน การกำหนดเกณฑ์ (threshold) ไม่จำเป็นจะต้องเป็น 0.7 เสมอไป สามารถกำหนดได้ตามความต้องการโดยต้องคำนึงถึงความสามารถที่จะครอบคลุมคำได้มาก (exhaustivity)

4.5 ความถี่ของ Neighborhood

เทคนิคการหาความถี่ของ neighborhood นับว่ามีประสิทธิภาพมากในการสร้างคำและกลุ่มคำ สามารถนำไปใช้กับการสร้างคำในภาษาอื่นได้ การกำหนดค่า (threshold) เพื่อนำ token มาเรียงเข้าด้วยกันก็สามารถปรับได้ตามความเหมาะสม ถ้าต้องการได้คำที่สั้นกระทัดรัดก็กำหนดค่า threshold ให้สูงขึ้น เทคนิคนี้จะมีข้อบกพร่องอยู่บ้างตรงวิธีการกำจัด term ที่ซ้ำซ้อน เพราะไม่อาจทราบได้ว่า คำแต่ละคำ เริ่มต้นหรือจบที่ token ตัวไหน เมื่อตัดบาง term ออกไป อาจมีผลทำให้ term ที่เหลืออยู่มีใจความไม่สมบูรณ์ ซึ่งปัญหานี้สามารถแก้ไขได้ในบางขั้นตอน แต่ต้องใช้ความพยายามอย่างมากในการวิเคราะห์คำไทย เพื่อสร้างกฎเกณฑ์ขึ้น

4.6 คำที่เป็น Stopword

การรวบรวม stopword ภาษาไทยเป็นเรื่องที่มีปัญหามาก คำใน stopword คำใดคำหนึ่งสามารถมีได้ทั้งคำที่ให้ความหมายและไม่ให้ความหมาย เมื่อใช้เป็น stopword ก็ทำให้เกิดปัญหาทำให้คำมีใจความไม่สมบูรณ์ (incomplete word) หากไม่ใช้เป็น stopword ก็ทำให้มีคำที่ไม่มีความสำคัญต่อเนื้อหาเอกสาร เพราะคำเหล่านี้ส่วนใหญ่จะมี frequency สูง ในทางตรงกันข้าม เมื่อถูกเรียงเข้ากับคำอื่น (concatenation) ก็จะทำให้กลุ่มคำมีขนาดใหญ่เกินไป (word size) ตัวอย่างคำที่เป็น stopword

stopword ก็**ดี**, **ที่**, **อยู่**, มิฉะนั้น, **หลัง**จาก (token คือคำที่พิมพ์ด้วยตัวหนา)

คำไทย **ที่**อยู่อาศัย, เจ้าหน้า**ที่**, **ดี**บุก, วี**ดี**โอคาสเซท, มันสำปะ**หลัง**, คอนโด**มิ**เนียม, **มิ**นิคอมพิวเตอร์

5. บทสรุป

กล่าวได้ว่า ระบบการทำดรรชนีคำสำคัญเป็นระบบที่สมควรแก่การสนับสนุนและส่งเสริมให้มีการค้นคว้าวิจัยกันอย่างจริงจังและกว้างขวางยิ่งขึ้น เพื่อให้การประมวลผลภาษาไทยตามลักษณะภาษาธรรมชาติได้พัฒนาอย่างมีประสิทธิผลและประสิทธิภาพ การทำดรรชนีคำสำคัญที่ได้กล่าวมานี้จะเป็นประโยชน์ต่อการสืบค้นสารนิเทศโดยตรง คือ ผู้ใช้ (user) สามารถทำการสืบค้นตามลักษณะภาษาธรรมชาติได้โดยประมวลคำสำคัญที่ใช้สืบค้นออกเป็น token เพื่อตรวจสอบกับ token ที่เป็นคำสำคัญ วิธีนี้ไม่ต้องอาศัยคู่มือ และรูปแบบคำถามในการสืบค้น ข้อเสนอแนะที่ควรทำต่อไปของงานวิจัยนี้คือ ศึกษาและปรับปรุงอัลกอริทึมในการแบ่งอักขระไทยยอกเป็น token เพื่อผลลัพธ์ที่ดียิ่งขึ้น ทดลองสืบค้นสารนิเทศตามลักษณะภาษาธรรมชาติ ศึกษาคำไทยที่ควรใช้เป็น stopword โดยเริ่มจากแต่ละสาขาวิชา ศึกษาคำไทยที่เป็นได้ทั้งคำที่มีความหมายและไม่มีความหมาย ดังนี้เป็นต้น

TF_1 → การศึกษาพฤติกรรมผู้บริโภค ที่มีต่อผลิตภัณฑ์แชมพูสระผมในเขตกรุงเทพมหานครนี้เป็นหัวข้อเรื่องที่น่าศึกษาเรื่องหนึ่ง เพราะแชมพูสระผมได้เข้ามามีบทบาทในชีวิตความเป็นอยู่ของคนไทยมาเป็นเวลาช้านานแล้ว แต่ยังไม่มีบุคคลใดทำการศึกษาค้นคว้ากันอย่างจริงจัง ขณะเดียวกันธุรกิจด้านแชมพูสระผมมีอัตราการขยายตัวสูงขึ้น มีตรายี่ห้อต่าง ๆ ที่จำหน่ายอยู่ในท้องตลาดกว่า 40 ตรายี่ห้อ ซึ่งต่างก็พยายามใช้กลยุทธ์ต่าง ๆ เพื่อช่วงชิง

ส่วนแบ่งตลาด (market share) ในตลาดแชมพู ด้วยเหตุนี้ผู้เขียนจึงได้ทำการศึกษาในหัวข้อดังกล่าว

TF_2 ⟶ การศึกษาพฤติกรรม ผู้บริโภค

เมื่อข้อความใน TF_1 แยกออกเป็น tokens

* ที่ มี ต่อ ผลิต ภัณฑ์ แชมพูสระ ผม ใน เขตกรุง เทพ-มหานคร นี้ เป็น หัว ข้อ เรื่อง ที่ น่า ศึกษา เรื่อง หนึ่ง

* เพราะ แชมพูสระ ผม ได้ เข้า มา มี บทบาท ใน ชี วิต ความ เป็นอยู่ของ คน ไทยมา เป็น เวลา ช้า นาน แล้ว

* แต่ ยัง ไม่ มี บุคคล ใด ทำ การศึกษา ค้นคว้า กันอย่าง จริง จัง

* ขณะ เดียว กัน

ธุรกิจด้าน แชมพูสระ ผมมี อัตรา การขยาย ตัวสูง ขึ้น

* มีตรา ยี่ ห้อต่าง

* ๆ

* ที่ จำ หน่ายอยู่ ในท้อง ตลาดกว่า

* 40

ตรา ยี่ ห้อ

* ซึ่ง ต่างก็ พยา ยาม ใช้กลยุทธ์ ต่าง

* ๆ

* เพื่อช่วง ชิง ส่วน แบ่ง ตลาด

* (market

* share)

* ในตลาด แชมพู

ด้วย เหตุ นี้ ผู้ เขียน จึง ได้ ทำ การศึกษา ใน หัว ข้อ ดังกล่าว

TF_3 ⟶ การศึกษาพฤติกรรม ผู้บริ โภค

tokens ที่เป็น stop-word ถูกตัดออก

* ผลิต ภัณฑ์ **แชมพูสระ ผม** * เขตกรุง เทพมหานคร

* หัว ข้อ * ศึกษา

* **แชมพูสระ ผม** * บทบาท * ชี วิต

* คน ไทยมา * เวลา

* บุคคล * การศึกษา ค้นคว้า
* ธุรกิจด้าน **แชมพูสระ ผมมี** อัตรา การขยาย ตัวสูง
* มีตรา ยี่ ห้อต่าง
* จำ หน่ายอยู่ * ตลาดกว่า
* ตรา ยี่ ห้อ
* พยา ยาม ใช้ กลยุทธ์
* เพื่อช่วง ชิง * แบ่ง ตลาด
* ในตลาด **แชมพู**
* ผู้ เขียน * การศึกษา * หัว ข้อ

เรื่องที่ 1 การศึกษาพฤติกรรมของผู้บริโภคที่มีต่อผลิตภัณฑ์แชมพูสระผมในเขตกรุงเทพมหานคร (A Study on Consumer's Behavior Towards Shampoo in Bangkok Metropolitan Area).

เรื่องที่ 2 ปัจจัยในการพิจารณาเลือกใช้กิจการตัวแทนโฆษณา (Factors Considered in Selecting the Advertising Agency).

เรื่องที่ 3 การศึกษาแนวโน้มของการประยุกต์ใช้แนวความคิดทางการตลาดในธุรกิจโรงแรม (A Study on the Trend of Applying Marketing Concepts to Hotel Business).

ตัวอย่างคำสำคัญอัตโนมัติจาก 3 เรื่อง ซึ่งประกอบด้วย **ความถี่ Tokens ที่อยู่ติดกันและคำสำคัญ** เรื่องละ 10 คำ

เรื่องที่ 1

24--→ 1 2--→ ผู้บริโภค
10--→ 4 9--→ ยี่ห้อ
8--→ 7 3--→ แชมพูสระผม
6--→ 5 10--→ จำหน่าย
6--→ 11 23--→ ผลิตภัณฑ์
6--→ 15 3--→ เส้นผม
6--→ 16 19--→ หนังศีรษะ
5--→ 6--→ โฆษณา
5--→ 8--→ แชมพู
4--→ 12--→ ทาง

เรื่องที่ 2

156--→ 4 2 3 1--→ กิจการตัวแทนโฆษณา
35--→ 1--→ โฆษณา
32--→ 2 3--→ ตัวแทน
30--→ 10 12 11--→ ผู้ประกอบธุรกิจ
28--→ 21 5 27 17--→ เป็นองค์ประกอบสำคัญ
16--→ 18 1--→ ผลิตงานโฆษณา
15--→ 6 22 9--→ ผู้วิจัย
14--→ 14 23--→ สามารถ
13--→ 5--→ ประ
13--→ 7--→ หน่วยธุรกิจ

เรื่องที่ 3

27--→ 4 1 2--→ ธุรกิจโรงแรม

15--→ 6 7 3--→ อุตสาหกรรมการท่องเที่ยว

10--→ 1 2--→ โรงแรม

10--→ 26 31 29 1 2--→ อัตราภาษีโรงแรม

9--→ 22 11 8--→ เสริมการลงทุน

9--→ 34 13 3--→ นักท่องเที่ยว

8--→ 1--→ โรง

8--→ 10 12--→ สามารถ

8--→ 14 23 1 2--→ ในการดำเนินธุรกิจโรงแรม

8--→ 15 9 1 2--→ เป็นอุตสาหกรรมโรงแรม

REFERENCE

Baxendale, P.B. (1958) "Machine-made index for technical literature and experiment." **IBM Journal for Research and Development.** 2(4) : 354 - 362.

Bookstein, A. and Swanson, D.R. (1975) "A decision-theoretic foundation for indexing." **Journal of the American Society for Information Science.** 26(1) : 45 - 50.

Borko, H. and Bernier, C.L. (1978) **Indexing concepts and method.** New York : Academic Press.

Boyce, B.R. and Kraft, D.M. (1985) "Principles and theories in information science." in **Annual Review of Information Science and Technology.** 20 : 154 - 178.

Carrol, J.M. and Roeloffs, R. (1969) "Computer selection of keywords using word-frequency analysis". **American Documentation.** 20(3) : 227 – 233.

Damerau, F.J. (1965) "An experiment in automatic indexing." **American Documentation.** 16(4) : 283 - 289.

Doszkocs, T.E. (1986) "Natural language processing in information retrieval." **Journal of the American Society for Information Science.** 37(4) : 191 - 196.

Foskett, A.C. (1982) **The Subject approach to information.** London : Clive Bingley.

Lancaster, F.W. (1979) **Information retrieval systems : characteristics, testing and evaluation.** New York : Wiley.

Luhn, H.P. (1957) "A statistical approach to mechanized encoding and searching of library information." **IBM Journal of Research and Development.** 1(4) : 309 - 317.

Pao, M.L. (1978) "Automatic text analysis based on transition phenomena of word occurrence." **Journal of the American Society for Information Science.** 29(3) : 121 - 124.

Ro, J.S. (1985) **An evaluation of applicability of ranking algorithm to improving the effectiveness of full text retrieval.** Ph.D. Dissertation Indiana University.

Salton G. (1969) "A comparison between manual and automatic indexing methods." **American Documentation.** 20(1) : 61 – 71.

Salton, G. and McGill, M.J. (1983) **Introduction to modern information retrieval.** New York : McGraw-Hill.

Shibayama, M. (1987) "Input/output methods for Thai-development of a database and a computer concordance for the Three Seals Law of Thailand." **Southeast Asia Studies.** 15(2) : 279 – 296.

Sparck Jones, K. (1973) "Indexing term weighting." **Information Storage and Retrieval** 9(11) : 619 - 633.

_____. (1974) "Progress in documentation : Automatic indexing." **Journal of Documentation.** 30(4) : 393 432.

Sparck Jones, K. (ed.) (1981) **Information retrieval experiment.** London : Butterworths.

Van Rijsbergen, C.J. (1979) **Information Retrieval.** 2 nd. ed. London : Butterworths.

Weinberg, B.H. (1981) **Word frequency and automatic indexing.** Ph.D. Dissertation, Columbia University.

วิชิต หล่อจีระชุณห์กุล และเจริญ คุวินทร์พันธุ์ (2524) **Thai-Soundex Algorithm and Thai-Syllable Separation Algorithm.** เอกสารเสนอต่อคณะกรรมการส่งเสริมงานวิจัย สถาบันบัณฑิตพัฒนบริหารศาสตร์.