บทที่ 3

การวิเคราะห์และการออกแบบ

สำหรับงานวิจัยนี้จะอาศัยโปรแกรมส่วนจัดเก็บดัชนีจากงานวิจัยเกี่ยวกับต้นไม้แพ็ต (เปรมิน จินดาวิมลเลิศ, 2539) เพื่อใช้จัดเก็บดัชนี และนำผลลัพธ์ดัชนีที่ได้มาพัฒนาโปรแกรมค้นคืนเพื่อขยายส่วนการค้นคืนเพื่อให้สามารถค้นคืนตามรูปแบบจัดลำดับ และแบบค้นคืนย้อนกลับ สำหรับระบบที่พัฒนาขึ้นเพิ่มเติมแบ่งออกเป็น 3 ส่วนดังนี้

1. การพัฒนาส่วนจัดเก็บดัชนีเพื่อเก็บค่าน้ำหนักคำ
2. การค้นคืน และจัดเก็บผลการค้นคืน
3. ขั้นตอนการค้นคืนย้อนกลับ

3.1 การพัฒนาส่วนจัดเก็บดัชนีเพื่อเก็บค่าน้ำหนักคำ

การพัฒนาส่วนจัดเก็บดัชนีแบบแถวลำดับแพ็ตทำได้โดยนำแฟ้มเอกสารทั้งหมด มาผ่านโปรแกรมเพื่อใช้สร้างต้นไม้แพ็ตและแปลงต้นไม้แพ็ตเป็นแถวลำดับแพ็ต แล้วจึงมีการสร้างแถวลำดับแพ็ตแบบสั้น เพื่อใช้ช่วยการค้นคืนให้รวดเร็วยิ่งขึ้น สำหรับรายละเอียดการสร้างดัชนีอ่านเพิ่มเติมได้ที่ (เปรมิน จินดาวิมลเลิศ, 2539)

ข้อมูลที่สำคัญกับระบบค้นคืนแบบจัดลำดับและค้นคืนย้อนกลับ คือค่าของน้ำหนักคำ ซึ่งเป็นค่าที่เกิดจากการคำนวณในช่วงของการค้นคืนซึ่งเป็นขั้นตอนที่มีการประมวลผลสูงมาก ซึ่งถ้าต้องการลดขั้นตอนการประมวลผล สามารถใช้วิธีเก็บค่าบางค่าที่เกิดจากการประมวลผลก่อน (Preprocessing) ซึ่งสามารถทำได้ 4 รูปแบบดังนี้

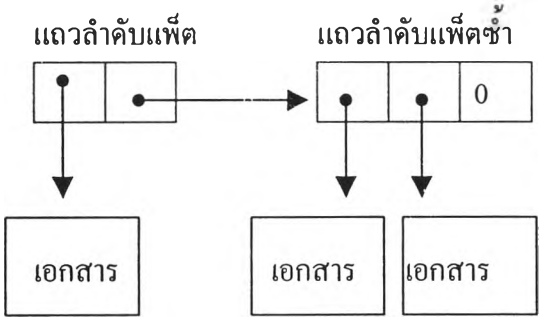
1. เก็บค่าความถี่ของคำแท้จริง วิธีนี้จะให้ผลการค้นคืนช้า แต่เป็นวิธีที่เหมาะสมที่สุดกับขั้นตอนการให้น้ำหนักคำในการวิจัยนี้คือสามารถนำค่าความถี่ของคำแท้จริงที่ได้ไปใช้กับสูตรคำนวณค่าน้ำหนักคำที่ต้องการได้ทันที

2. เก็บค่าความถี่ของคำแบบนอร์มัลไลเซชัน ซึ่งอาจเป็นค่าความถี่ของคำแบบล็อคการิทึม หรือค่าความถี่ของคำแบบขยาย เป็นวิธีที่ให้ผลการค้นคืนเร็วกว่าวิธีที่ 1 แต่วิธีนี้ไม่เหมาะกับการคำนวณของสูตรน้ำหนักคำที่ต้องการใช้ค่าความถี่ของคำแท้จริง

3. เก็บค่าความถี่ของคำแบบสมบูรณ์ โดยการคำนวณผลน้ำหนักคำไว้ก่อน ซึ่งให้ผลการค้นคืนเร็วที่สุด แต่ไม่เหมาะกับระบบที่มีการเปลี่ยนแปลงรูปแบบการให้น้ำหนักคำ

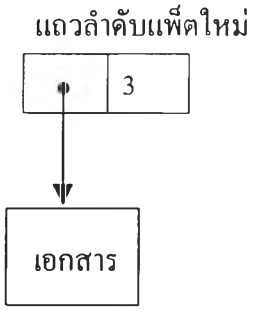
4. ไม่มีเก็บค่าน้ำหนักคำ การคำนวณน้ำหนักคำจะทำในขั้นตอนการค้นคืนทั้งหมดซึ่งทำให้ช่วงการค้นคืนต้องมีการประมวลผลสูง ทำให้ระบบช้า

สำหรับในที่นี้แถวลำดับแพ็ตคือแถวลำดับของตัวชี้ ประกอบด้วยตัวชี้ตำแหน่งซิสตริงในเอกสารและตัวชี้ไปยังแถวลำดับแพ็ตซ้ำ โดยที่ตัวชี้จะชี้ไปยังตำแหน่งซิสตริงในเอกสาร แต่ละรายการของตัวชี้จะถูกจัดเรียงตามลำดับค่าแอสกีของทุกๆซิสตริงที่มันชี้อยู่ในเอกสาร ส่วนตัวชี้แถวลำดับแพ็ตซ้ำจะชี้ไปยังแถวลำดับแพ็ตซ้ำ ที่เก็บตำแหน่งของค่าซิสตริงที่ซ้ำกัน โดยถ้าพบว่าค่าตำแหน่งซิสตริงซ้ำมีค่าเป็น 0 แสดงว่าหมดซิสตริงซ้ำตัวนั้น ดังรูปที่ 3.1



รูปที่ 3.1 แสดงโครงสร้างแถวลำดับแพ็ต

จากโครงสร้างแถวลำดับแพ็ตข้างต้นยังไม่มีข้อมูลที่จำเป็นเพื่อใช้คำนวณน้ำหนักคำเก็บไว้ ดังนั้นจำเป็นต้องมีการเตรียมค่าบางอย่าง ซึ่งในที่นี้เลือกเก็บค่าความถี่แท้จริงไว้ล่วงหน้า เพื่อให้สามารถนำค่าความถี่แท้จริงนี้ไปใช้คำนวณตามสูตรหาน้ำหนักคำได้ต่อไป โดยมีโครงสร้างดังรูปที่ 3.2 ซึ่งอาศัยข้อมูลจากแถวลำดับแพ็ตข้างต้น มาใช้ค้นคืนเพื่อหาความถี่ของคำของแต่ละซิสตริงที่อยู่ในแถวลำดับ โดยผลที่ได้จะเป็นแฟ้มดัชนีใหม่ที่มีข้อมูลความถี่ของคำของแต่ละซิสตริงเก็บไว้ด้วย จากรูปที่ 3.2 ค่า 3 เป็นจำนวนซิสตริงทั้งหมดหรือความถี่ของซิสตริงที่ได้จากรูปที่ 3.1ซึ่งทำให้ลดเวลาประมวลผลในช่วงการค้นคืนได้



รูปที่ 3.2 แสดงโครงสร้างแถวลำดับแพ็ตใหม่

โครงสร้างดัชนีใหม่ที่มีข้อมูลความถี่ซิสตริงเก็บไว้ จะอาศัยโครงสร้างข้อมูลจากดัชนีเดิม โดยโครงสร้างแฟ้มข้อมูลความถี่ซิสตริง ประกอบด้วย 2 ส่วนคือ

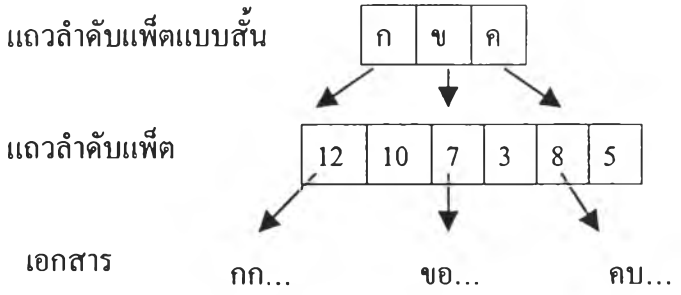
1 ส่วนหัว (Header) มีองค์ประกอบหลักของข้อมูลเหมือนแถวลำดับแพ็ตเดิม คือ
   - ชื่อแฟ้มแถวลำดับแพ็ตซ้ำ
   - ชื่อแฟ้มตารางการผสานแถวลำดับแพ็ตแถวลำดับแพ็ต
   - จำนวนแถวลำดับแพ็ตทั้งหมด
   - จำนวนแถวลำดับแพ็ตซ้ำทั้งหมด
   - จำนวนแถวลำดับแพ็ตซ้ำในตอนผสานแถวลำดับแพ็ต
   - ความยาวซิสตริง
   - จำนวนชิ้นที่ทำการผสานแถวลำดับแพ็ต
   - ตัวระบุแฟ้มต้นฉบับย่อยเป็นฉบับสุดท้าย
2 ส่วนข้อมูล (Detail) เก็บข้อมูลดังนี้
   - ตำแหน่งซิสตริงที่ใช้ในการอ้างอิงถึงซิสตริงในแฟ้มต้นฉบับ
   - ความถี่ของซิสตริงที่พบในเอกสาร

ดังนั้นในระบบค้นคืนมีแฟ้มข้อมูลที่ใช้ในระบบดังนี้

1. แฟ้มข้อมูลเก็บเอกสาร เป็นแฟ้มที่สร้างขึ้นเพื่อใช้เก็บข้อความต่าง ๆ
2. แฟ้มข้อมูลดัชนี เป็นแฟ้มที่เก็บข้อมูลเกี่ยวกับแถวลำดับแพ็ต เพื่อใช้ในการค้นคืน
3. แฟ้มข้อมูลน้ำหนักคำ เป็นแฟ้มที่สร้างขึ้นใหม่ เพื่อใช้ในการคำนวณหาค่าน้ำหนักคำ

3.2 การค้นคืนและจัดเก็บผลการค้นคืน

ขั้นตอนการค้นคืนบนแถวลำดับแพ็ต ใช้วิธีค้นหาแบบทวิภาค (Binary search) แต่เนื่องจากขนาดแถวลำดับแพ็ตมักมีขนาดใหญ่ เกินกว่าจะเก็บไว้ได้หมดในหน่วยความจำ ทำให้ต้องทำการค้นหาแบบทวิภาคบนดิสก์ ซึ่งทำให้ประสิทธิภาพของระบบลดลง จึงได้มีการสร้างแถวลำดับแพ็ตแบบสั้น เพื่อคลุมแถวลำดับแพ็ตอีกทอดหนึ่ง โดยแถวลำดับแพ็ตแบบสั้นจะเก็บตัวชี้ไปยังที่เริ่มตัวอักษรนั้นบนแถวลำดับแพ็ต ดังรูปที่ 3.3 ทำให้การค้นคืนไม่ต้องค้นหาแบบทวิภาคทั้งแถวลำดับแพ็ต แต่จะค้นหาแบบทวิภาคเพียงบางส่วนเท่านั้น เช่น การค้นหาเฉพาะช่วง คำที่ขึ้นต้นด้วย "ก" (คล้ายกับการค้นหาชื่อคนในสมุดโทรศัพท์) โดยค้นหาแบบทวิภาคบนแถวลำดับแพ็ตแบบสั้นก่อน จะได้ค่าตำแหน่งของข้อมูลที่ขึ้นต้นด้วย "ก" บนแถวลำดับแพ็ต แล้วจึงโหลดช่วงข้อมูลนั้นๆ เข้าสู่หน่วยความจำเพื่อทำการค้นหาแบบทวิภาค ก็จะได้ข้อมูลตามต้องการ

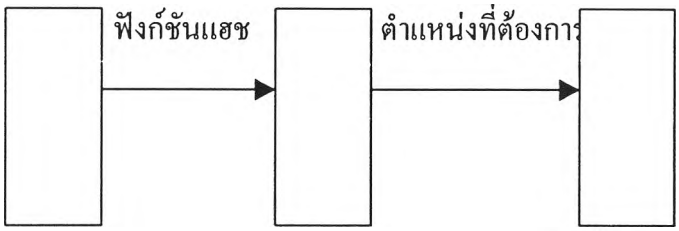


รูปที่ 3.3 แสดงการค้นคืนบนแถวลำดับแพ็ต

การค้นคืนเพื่อสร้างผลลัพธ์ที่เรียงลำดับ สิ่งแรกคือต้องค้นคืนเอกสารเพื่อหาคำที่ต้องการ แล้วจะได้ผลลัพธ์เป็นรายการของเอกสารที่มีคำที่ต้องการและจำนวนของคำที่ต้องการปรากฏในเอกสารนั้นๆ จำนวนของคำที่ปรากฏในเอกสารเป็นข้อมูลความถี่ของแต่ละคำในเอกสาร ซึ่งจะนำมาคำนวณหาค่าน้ำหนักรวมของแต่ละเอกสาร และสุดท้ายจึงนำค่าน้ำหนักที่ได้มาจัดเรียงลำดับ จะเห็นว่าขั้นตอนนี้เวลาที่ใช้ ขึ้นกับจำนวนเอกสารที่ค้นคืนได้ เนื่องจากการเก็บข้อมูลของผลลัพธ์การค้นคืน ต้องใช้เวลาหาตำแหน่งเพื่อเก็บน้ำหนักคำของเอกสารหนึ่งๆ ซึ่งอาจใช้การค้นหาแบบทวิภาค ซึ่งมีประสิทธิภาพ O(log(n))

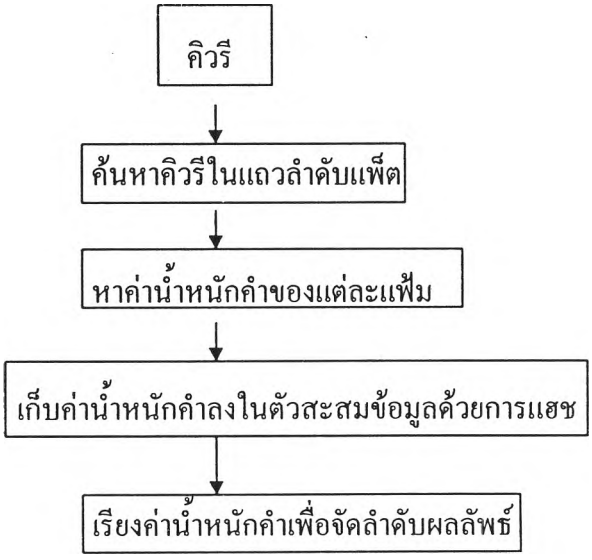
เราสามารถลดเวลาที่ใช้กับขั้นตอนนี้โดยใช้วิธีการแฮช(Hashing) ซึ่งเป็นโครงสร้างที่เหมาะกับการค้นหา และ การเพิ่ม ที่ให้ประสิทธิภาพ O(1) โดยใช้ตารางแฮชแทนตัวสะสมข้อมูล (Accumulator) เมื่อได้ผลการค้นคืน จะต้องมีการคำนวณค่าน้ำหนักคำของเอกสารโดยทำการคำนวณค่าฟังก์ชันแฮช เพื่อเข้าถึงตำแหน่งน้ำหนักคำของเอกสาร ฟังก์ชันแฮชคำนวณหาค่าคีย์โดย

อาศัยชื่อเอกสารซึ่งไม่มีซ้ำ มาคำนวณค่าตามรหัสแอสกีแล้วหารแบบเอาเศษกับค่าจำนวนเอกสารทั้งหมดจะได้ผลลัพธ์เป็นค่าคีย์ และจองขนาดตารางแฮช เท่ากับจำนวนเอกสารดังรูปที่ 3.4 สำหรับปัญหาการชนกันของค่าคีย์ จะใช้วิธีแก้ปัญหาเชิงเส้น (Linear probing) เป็นการนำค่าคีย์ที่ชนไปยังตำแหน่งว่างถัดไป



รูปที่ 3.4 แสดงการแฮชเพื่อหาตำแหน่งเก็บข้อมูล

เมื่อได้ผลลัพธ์จากวิธีการข้างต้นแล้วจึงนำค่าจากตารางแฮช ไปผ่านขั้นตอนการเรียงลำดับต่อไป การแฮชทำให้ขั้นตอนการค้นหาเอกสารขั้นนี้ไม่ขึ้นกับจำนวนเอกสารที่ค้นคืนแต่ยังคงมีขั้นตอนเรียงลำดับซึ่งเป็นขั้นสุดท้ายของการจัดลำดับเอกสารยังคงขึ้นกับจำนวนเอกสารที่ค้นคืนมาได้ สำหรับขั้นตอนทั้งหมดแสดงได้ดังรูปที่ 3.5

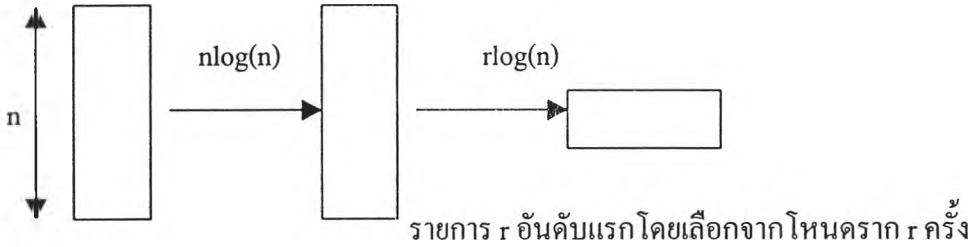


รูปที่ 3.5 แสดงขั้นตอนการค้นคืนข้อมูลแบบจัดลำดับ

การค้นหาดิวรีในแถวลำดับแพ็ต เนื่องจากข้อมูลในแถวลำดับเป็นรูปแบบของซิสตริง ดังนั้นต้องมีการตรวจสอบว่าตำแหน่งซิสตริงที่ได้เป็นคำที่ต้องการจริงๆ และเป็นคำที่ถูกต้องตามหลัก

ภาษาศาสตร์ ไม่ได้เป็นเพียงส่วนหนึ่งของคำเช่น ถ้าต้องการหาคำว่า “ink” เมื่อค้นคืนและหาค่าความถี่ของคำ เราจะนับค่าความถี่ของคำเฉพาะคำที่พบว่าเป็น “ink” จริงโดยไม่รวม “ink” ที่เกิดจากคำอื่นๆ เช่น “link” หรือ “inkling” ซึ่งในภาษาอังกฤษสามารถตรวจสอบว่าซิสตริงที่ได้เป็นคำว่า “ink” จริง โดยการตรวจสอบตัวอักษรก่อนและหลัง “ink” ว่าเป็นตัวว่าง (Blank) หรือตัวแบ่งคำอื่นๆ หรือไม่ โดยถ้าเป็นตัวแบ่งคำจึงจะแสดงว่าเป็นคำที่ต้องการจริง แต่ในภาษาไทยซึ่งคำแต่ละคำจะอยู่ติดกันได้ ไม่สามารถตรวจสอบตัวแบ่งคำได้เหมือนภาษาอังกฤษเช่น ถ้าต้องการหาคำว่า “กร” ต้องมีการตรวจสอบว่าเป็นคำว่า “กร” จริง โดยไม่รวมคำว่า “กรณ์” และ “กรม” เข้าไปด้วย ดังนั้นจึงต้องมีการแบ่งคำในซิสตริงก่อนซึ่งในการวิจัยนี้เลือกใช้ระบบแบ่งคำในระบบปฏิบัติการวินโดว์ภาษาไทยมาใช้แบ่งคำแล้วจึงตรวจสอบคำที่ถูกแบ่งคำแล้วว่าเป็นตรงกับคำที่ต้องการหรือไม่ ถ้าตรงกับคำที่ต้องการจริง แล้วค่อยนับค่าความถี่ของคำนั้นๆ ต่อไป วิธีนี้เป็นการตรวจสอบเบื้องต้นโดยการแบ่งคำสามารถปรับปรุงให้มีความถูกต้องและเหมาะสมมากขึ้นภายหลังได้

ส่วนการเรียงค่าน้ำหนักคำในการวิจัยนี้จะใช้โครงสร้างฮีพช่วยในการจัดลำดับผลการค้นคืนจากคำที่มีน้ำหนักคำมากไปยังคำที่มีน้ำหนักคำน้อย โดยมีขั้นตอนดังรูปที่ 3.6



รูปที่ 3.6 แสดงขั้นตอนการเรียงลำดับข้อมูลด้วยฮีพ

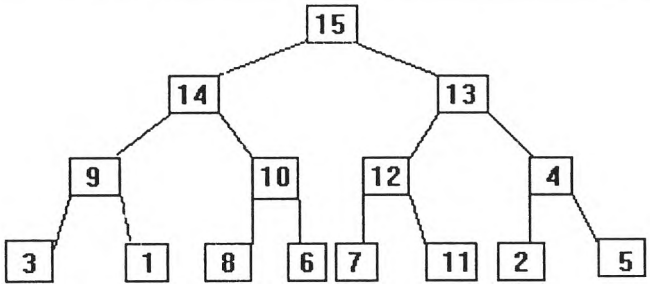
เริ่มต้นจะนำค่าผลลัพธ์ที่ได้จากการค้นคืน ซึ่งอยู่ในรูปของแถวลำดับ (ชื่อเอกสาร จำนวนที่พบและน้ำหนักคำ) ที่มีค่าน้ำหนักคำมากกว่าศูนย์ มาใส่ลงในแถวลำดับฮีพ แล้วจึงเริ่มขั้นตอนสร้างฮีพแบบค่าสูงสุด เรียงตามค่าน้ำหนักคำ โดยใช้เวลา O(nlog(n)) แล้วจึงดึงค่าโหนดราก มาใส่ในผลลัพธ์ที่เรียงแล้วจำนวน r ครั้ง (r เป็นจำนวนรายการที่ต้องการให้แสดงต่อผู้ใช้ เช่นแสดงเฉพาะ 10 เอกสารแรกที่ค้นเจอ) ซึ่งขั้นตอนนี้เมื่อดึงโหนดรากไปแล้ว ต้องมีการปรับฮีพ โดยใช้เวลา O(rlog(n))

การเรียงลำดับแบบฮีพหรือบางครั้งเรียกว่าการเรียงลำดับแบบต้นไม้ เป็นการเรียงข้อมูลที่อาศัยโครงสร้างต้นไม้ไบนารี โครงสร้างนี้จะต้องมีคุณสมบัติว่า โหนดใดๆในต้นไม้นั้นจะต้องมีค่าคีย์ใหญ่กว่าคีย์ที่อยู่ในโหนดลูกของมัน จากโครงสร้างฮีพ จะเห็นว่าโหนดราก (root node) จะต้องมีค่าโหนดที่มีค่าใหญ่กว่าทุกๆโหนด เพราะฉะนั้นถ้าต้องการคีย์ที่มีค่ามากที่สุด จะได้โหนดรากเป็นคำตอบ

เราสามารถแทนที่ต้นไม้ฮีพด้วยแถวลำดับได้ โดยที่โหนด i ใดๆ สามารถไปถึงโหนดลูกซ้ายขวา คือไปยังตำแหน่ง 2*i และ 2*i + 1

ตัวอย่างโครงสร้างฮีพดังรูปที่ 3.7 จะเห็นว่าที่โหนดรากจะเก็บค่าที่มากที่สุดของข้อมูล

Index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Value	15	14	13	9	10	12	4	3	1	8	6	7	11	2	5



รูปที่ 3.7 แสดงโครงสร้างฮีพ

ประสิทธิภาพของการเรียงลำดับข้อมูลแบบฮีพ นี้จะให้ประสิทธิภาพไม่ดีมากนัก ในกรณีที่มีจำนวนข้อมูลน้อย แต่ในกรณีที่มีจำนวนข้อมูลมาก จะเป็นวิธีที่มีประสิทธิภาพดีอีกวิธีหนึ่ง สำหรับข้อมูล n ตัว การเรียงลำดับแบบฮีพทำได้เสร็จในเวลา O(nlog(n)) เสมอ และเป็นวิธีการที่ต้องการใช้พื้นที่เพิ่มเติมน้อยที่สุด

จากการวิเคราะห์ชุดข้อมูลที่เข้ามาจัดเรียง ในกรณีเฉลี่ย และกรณีเลวที่สุด การเรียงลำดับแบบฮีพจะใช้เวลาไม่แตกต่างกันมากนักในทั้งสองกรณี ถ้าพิจารณากรณีข้อมูลเลวที่สุดแล้ว วิธีการเรียงลำดับแบบฮีพจะให้ผลดีกว่า วิธีจัดเรียงลำดับแบบเร็ว โดยสรุปแล้ว การเรียงลำดับแบบฮีพเหมาะสำหรับ การทำงานที่รับประกัน ประสิทธิภาพ คือทำงานได้ดีสม่ำเสมอกันตลอด โดย การเรียงลำดับแบบฮีพสามารถหลีกเลี่ยงเหตุการณ์ที่เกิดการด้อยประสิทธิภาพลงอย่างรุนแรงได้

3.3 ขั้นตอนการค้นคืนย้อนกลับ

จากผลการค้นคืนแบบจัดลำดับข้างต้น ถ้าผลลัพธ์ของการค้นคืนที่ได้ยังไม่เป็นที่พอใจต่อผู้ใช้ ให้ผู้ใช้แสดงรายการเอกสารที่ตรงตามต้องการมาเป็นตัวอย่าง เพื่อสร้างคิวรีใหม่ที่มีผลลัพธ์ของการค้นคืนที่ดีขึ้น ซึ่งขั้นตอนการค้นคืนย้อนกลับในการวิจัยนี้ประกอบด้วยขั้นตอนดังนี้

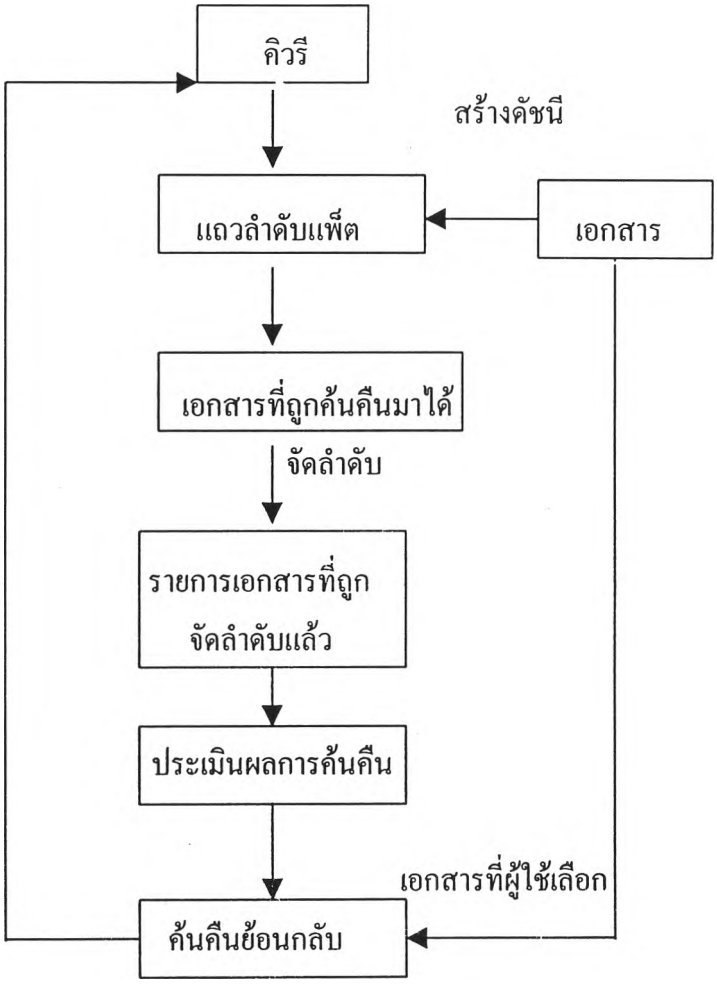
1. ผู้ใช้เลือกเอกสารโดยทำเครื่องหมายหน้ารายการเอกสารที่ตรงตามต้องการ ของผลลัพธ์การค้นคืนเริ่มต้น เป็นข้อมูลเข้า โดยเลือกได้มากกว่า 1 เอกสาร

2. นำคำทั้งหมดในเอกสารที่ป้อนมาแบ่งคำ ด้วยฟังก์ชันการแบ่งคำของระบบปฏิบัติการวินโดว์ภาษาไทย แล้วจึงนำคำทั้งหมดที่ได้จากการแบ่งคำนำมาจัดเรียงตามลำดับอักษร แล้วนำมาจัดเก็บแบบโครงสร้างแถวลำดับของคำและจำนวนเอกสารซึ่งตรงตามต้องการที่พบคำนั้น

3. นำคำแต่ละคำมาค้นคืนบนแถวลำดับแพ็ต เพื่อหาค่าความถี่ของคำ โดยถ้าคำใดที่มีความถี่ของคำเกินค่าขีดจำกัดบน หรือ ความถี่ของคำน้อยกว่าขีดจำกัดล่าง ก็จะตัดคำนั้นออก ไม่นำมาคำนวณน้ำหนักคำตามสูตรที่กล่าวไว้ในบทที่แล้ว

4. แสดงรายการน้ำหนักคำของแต่ละคำที่คำนวณได้ โดยเรียงตามลำดับน้ำหนักคำที่ได้จากมากไปน้อย เพื่อให้ผู้ใช้เลือกคำจากรายการคำเกี่ยวข้องที่แสดงนี้ไปเพิ่มในคิวรี เพื่อเริ่มค้นคืนใหม่อีกครั้งหนึ่ง

รูปที่ 3.8 แสดงขั้นตอนการทำงานทั้งหมดของระบบโดยสรุป ซึ่งประกอบด้วยการค้นคืนแบบจัดลำดับและการค้นคืนแบบค้นคืนย้อนกลับ เริ่มต้นที่รับคิวรีจากผู้ใช้แล้ว นำคิวรีไปค้นคืนในดัชนี จะได้ผลการค้นคืนเป็นรายการเอกสารที่มีคำที่ตรงกับคิวรี แล้วจึงนำค่าความถี่ของคำในเอกสารเหล่านั้นมาคำนวณหาค่าน้ำหนักคำ เมื่อได้ค่าน้ำหนักคำของแต่ละเอกสารแล้ว จะนำค่าน้ำหนักคำมาเรียงลำดับจากมากไปน้อย แล้วจึงแสดงผลให้ผู้ใช้เลือก และถ้าผลการค้นคืนที่ได้ ผู้ใช้ไม่พอใจ ผู้ใช้สามารถเลือกชุดเอกสารที่ผู้ใช้คิดว่าตรงตามต้องการป้อนกลับเข้าสู่ระบบ แล้วระบบจะนำเอกสารที่ผู้ใช้ป้อนกลับมา ทำการคำนวณหาน้ำหนักคำ เพื่อใช้เสนอคำหรือกลุ่มคำใหม่ให้ผู้ใช้เลือกเพื่อที่จะใช้สร้างคิวรีใหม่ที่ทำให้ผลการค้นคืนใหม่ที่ได้ดีขึ้น



รูปที่ 3.8 แสดงขั้นตอนการทำงานของระบบ