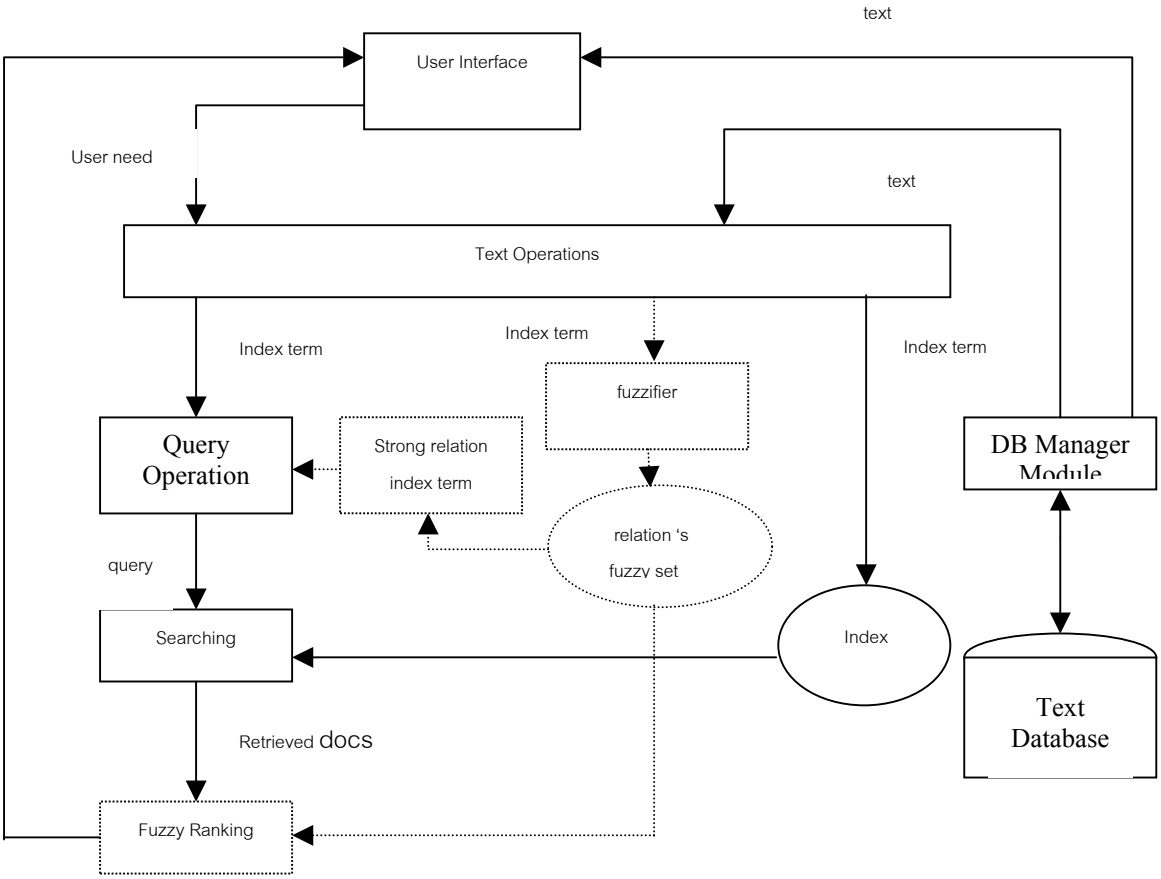
บทที่ 7

การออกแบบระบบ

7.1 องค์ประกอบของระบบ



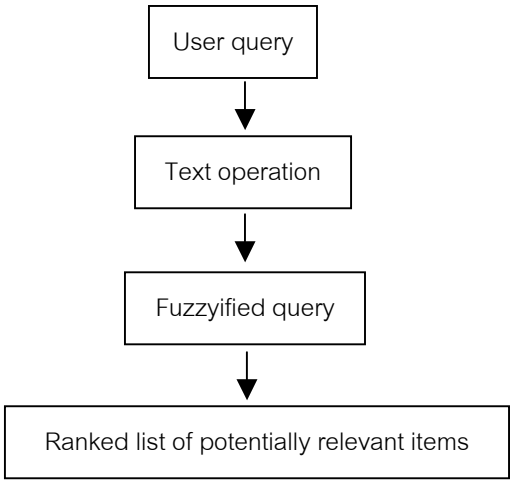
รูปที่ 7-1 ระบบการค้นคืนโดยภาพรวม

จากรูปที่ 7-1 ได้นำระบบการค้นคืนสารสนเทศที่ได้กล่าวไว้ในบทที่ 2 มาขยายประสิทธิภาพ โดยจากรูปจะนำเอาโครงสร้าง ของ ฟัซซี่ มาประกอบเข้าไปกับระบบเดิม

เราสามารถแบ่ง ระบบค้นคืนออกเป็นระบบ ย่อย ได้ 3 ส่วนด้วยกันคือ

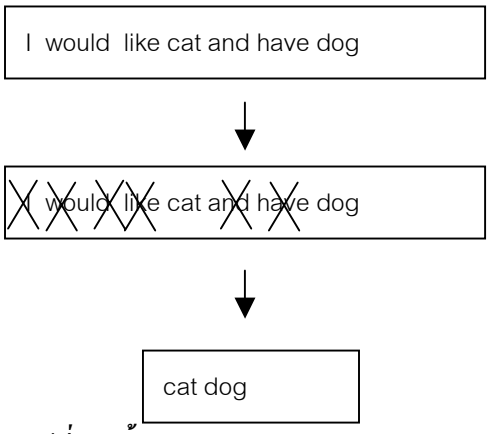
1. ส่วนการค้นหาเอกสารโดยใช้ ทฤษฎี ฟัซซี่ลอจิก
2. ส่วนการทำ index อัตโนมัติ
3. ส่วนการแก้ไขข้อมูลที่เกิดจากบรรณารักษ์

7.2 การค้นหาเอกสารโดยใช้ ทฤษฎี ฟัซซี่ลอจิก



รูปที่ 7-2 ขั้นตอนการค้นหาเอกสารโดยใช้ ทฤษฎี ฟัซซี่ลอจิก

จากรูป 7-2 ซึ่งแสดงขั้นตอนการค้นหา เอกสาร โดยใช้หลักการของ ฟัซซี่ ลอจิก โดยในขั้นแรกนั้น โปรแกรมจะรับความต้องการของผู้ใช้โดยผู้ใช้จะส่ง query เข้ามา โดย คำที่ผู้ใช้ต้องการค้นหา อาจมาในรูปแบบภาษาธรรมชาติ ได้ โดย โปรแกรมจะทำหน้าที่ช่วยการจัดโครงสร้างของภาษาเสียใหม่ ในส่วนของ Text Operations ยกตัวอย่างเช่น เมื่อผู้ใช้ใส่ query เข้ามาเป็น “I would like papers with have program and JSP” ในขั้นตอนการทำ Text operations จะทำการกำจัด คำที่ไม่ต้องการทิ้งออกไปให้เหลือแต่ คำที่ใช้ในการ query ได้



รูปที่ 7.3 ขั้นตอนการทำ Text Operation

เมื่อได้ index term ที่สามารถใช้ในการค้นหาได้แล้วในที่นี้คือ “cat” และ “dog” และทำการแยกคิด index term แต่ละตัว ในขั้นแรกจะนำ index term “cat” มาคิดก่อนโดยจะนำ index term “cat” ไปหาดูว่าในเอกสารไหนบ้างที่มีคำว่า “cat” ในตาราง ซึ่งการจัดเก็บเป็นดังนี้

Keyword	PNumber
cat	C0001
cat	C0078
dog	C0080
…	…
…	…

จากตัวอย่าง จะได้ เอกสารที่มี “cat” เป็น index term คือ C0001 และ C0078 หมายถึงเป็นเอกสารที่มี index term “cat” เป็นสมาชิกอย่างเต็มตัว ดังนั้นเอกสารเหล่านี้จะมีค่าความเป็นสมาชิกกับ index term “cat” เท่ากับ 1 แล้วนำ index term “cat” ไปหาคำเหมือน (Thesaurus) ซึ่งindex termที่เป็นคำเหมือนกับ index term “cat” จะถูกนำไปค้นหาด้วย แต่ index term เหล่านี้ จะมีค่าความเป็นสมาชิกไม่เท่ากับ index term “cat”

จากนั้นนำ index term “cat” ไปหาว่ามีความสัมพันธ์กับindex term อื่น ๆ อีกอะไรบ้างจากสูตรในการหาความสัมพันธ์ระหว่างindex termดังนี้

$$c_{i,j} = \frac{n_{i,j}}{n_i + n_j - n_{i,j}}$$

โดยจากสูตรสามารถอธิบายได้ดังนี้

$c_{i,j}$ คือ ความสัมพันธ์คู่index term i กับ j

$n_{i,j}$ คือ จำนวนเอกสารที่มีทั้ง index term i และj เป็นindex term

n_i คือ จำนวนเอกสารที่มี index term i เป็นindex term

n_j คือ จำนวนเอกสารที่มี index term j เป็นindex term

แต่ในทางปฏิบัติ ถ้าต้องทำการคำนวณความสัมพันธ์ทุก ๆ ครั้งที่มีการค้นหาจะทำเสียเวลาเป็นอย่างมากในการค้นคืนข้อมูลในผู้ใช้ ดังนั้น จะทำการจัดเก็บข้อมูลความสัมพันธ์ของ index termไว้ล่วงหน้าเพื่อความรวดเร็วในการค้นคืนข้อมูล ในการคิดคำนวณหาความสัมพันธ์ของindex termนั้นจะกระทำในขั้นตอนการใส่ข้อมูลของ บรรณารักษ์ หรือ การแล้วไขข้อมูลที่เกิดจาก บรรณารักษ์ หรือ ผู้ดูแลระบบเท่านั้น โดยการคำนวณทุก ๆ ครั้งจะคำนวณเฉพาะ index term ที่มีการเปลี่ยนแปลงเท่านั้น เพื่อใช้เวลาน้อยที่สุดในการคำนวณ

ตัวอย่างตารางความสัมพันธ์ของindex term

Index 1	Index 2	Relate
Cat	kitty	0.5
Cat	dog	0.25
Dog	puppy	0.5

เมื่อต้องการค้นหา index term คำว่า “cat” เราจะพบว่า index term “kitty” และ index term “dog” มีความสัมพันธกับindex term “cat” ดังนั้นต้องใช้index term “kitty” และ index term “dog” มาใช้ในการค้นหาเอกสารด้วย แต่ความเป็นสมาชิกของเอกสารที่มีต่อindex term “cat” จะไม่เท่ากับ 1 โดยจะขึ้นอยู่กับ คู่ความสัมพันธ์นั้น ๆ เช่น index term “cat” กับindex term “kitty” คือ 0.5 ดังนั้นเอกสารที่มี index term “kitty” ประกอบอยู่จะมีความเป็นสมาชิกต่อindex term “cat” เท่ากับ 0.5 และ เอกสารที่มี index term “dog” ประกอบอยู่ จะมีความเป็นสมาชิกต่อindex term “cat” เท่ากับ 0.25 แล้วนำไปเก็บไว้เพื่อรอการคำนวณ

ต่อมาก็ใช้ index term “dog” ในการค้นหาในทำนองเดียวกับการค้นหาโดยใช้ index term “cat” ในการค้นหาเมื่อได้ทำการเลือกเอกสารและกำหนดความสัมพันธ์ของเอกสารแล้วก็นำไปคำนวณโดยใช้สูตร

$$\mu_{i,j} = 1 - \prod_{k_l \in d_j} (1 - c_{i,l})$$

$\mu_{i,j}$ ความเป็นสมาชิกของเอกสารกับindex term

$\prod_{k_l \in d_j} (1 - c_{i,l})$ อินเวิด ของค่าความเป็นสมาชิกรวมในอินเด็กซ์ทุก ๆ เทอม

และเมื่อทำคำนวณเสร็จแล้วผลที่ได้ คือ ค่าความเป็นสมาชิกรวมของทุก ๆ อินเด็กซ์ที่มีผลต่อเอกสารนั้น ๆ โดยindex termหลัก คือ index term i เมื่อมีindex termหลาย ๆ ตัวเป็นตัว query จะต้องคำนวณตาม ลอจิก อีกครั้งเพื่อเป็นผลรวมของความสัมพันธ์ของ index termทุก ๆ ตัวที่เป็นตัว query ในการค้นคืนต่อเอกสารแต่ละฉบับโดยมีสูตรในการคำนวณดังต่อไปนี้

การกระทำ ลอจิก แบบ อินเตอร์เซ็ก

$$\mu_{first \cap president \cap u.s.a}(x) = \mu_{first}(x).\mu_{president}(x).\mu_{u.s.a}(x) \qquad \forall x \in U$$

การกระทำ ลอจิก แบบ ยูเนี่ยน

$$\mu_{first \cup president \cup u.s.a}(x) = (1-(1-\mu_{first}(x)).(1-\mu_{president}(x)).(1-\mu_{u.s.a}(x))) \qquad \forall x \in U$$

การกระทำ ลอจิก แบบ คอมพรีเมนต์

$$\bar{\mu}_{first}(x) = (1 - \mu_{first}(x)) \qquad \forall x \in U$$

เมื่อคำนวณตามรูปแบบเบื้องต้นแล้ว จะได้ผลรับที่ถูกเก็บไว้ในตัวแปรซึ่งมีรูปแบบดังนี้

Paper Number	Relation
C00001	1
C25789	0.667
R54873	0.873
R6579	0.248

เมื่อได้ข้อมูลออกมาแล้วนั้นยังไม่สามารถนำมาแสดงผลได้เพราะจำเป็นต้องทำการเรียงลำดับเอกสาร ที่มีความเป็นสมาชิกสูง ๆ ออกมาแสดงก่อน โดยต้องอาศัย การเรียงข้อมูลจากมากไปหาน้อย ในระบบค้นคืนข้อมูลโดยอาศัยทฤษฎีฟัซซี่ลอจิก นี้ ได้ใช้การเรียงข้อมูลแบบ Quick Sort

การเรียงลำดับข้อมูลโดยใช้ Quick Sort Algorithm

เนื่องจากการค้นคืนข้อมูลความสำคัญนอกเหนือจากความถูกต้องแล้ว ส่วนที่สำคัญมากอีกประการหนึ่งคือความเร็วในการค้นคืน การเรียงลำดับข้อมูลเป็นปัจจัยที่สำคัญยิ่งต่อการเพิ่มประสิทธิภาพของเรื่องความเร็ว โดยระบบค้นคืนโดยอาศัยทฤษฎีฟัซซี่ลอจิกนี้ ได้ใช้การเรียงข้อมูลแบบ Quick Sort Algorithm โดยคุณสมบัติของ Quick Sort Algorithm คือด้านความเร็วในการเรียงข้อมูล ซึ่งความซับซ้อนจะเท่ากับ O(n log n) ซึ่งจะเร็วกว่าการเรียงข้อมูลแบบอื่น ๆ เช่นการเรียงข้อมูลแบบ Bubble Sort Algorithm ซึ่งความซับซ้อนจะสูงกว่าคือ O(n^2) ซึ่งจะทำให้ประสิทธิภาพเรื่องความเร็วลดต่ำลง

เมื่อทำการเรียงลำดับเอกสารเสร็จเรียบร้อยแล้ว ข้อมูลก็อยู่ในรูปที่สามารถนำเสนอต่อผู้ค้นข้อมูลได้

7.3 การทำ index อัตโนมัติ

การทำ index อัตโนมัตินั้น แบ่งออกได้เป็น 2 ขั้นตอนการทำงาน ได้แก่

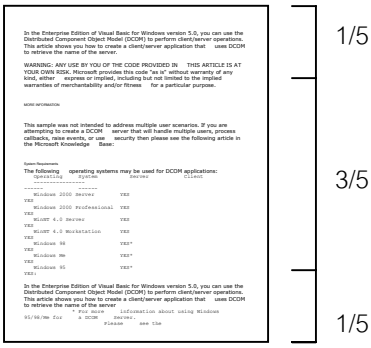
1) การเก็บสถิติความถี่ของคำต่างๆจากเอกสาร
2) การวิเคราะห์ข้อมูลด้วย Neural Network

การเก็บสถิติของคำจากแฟ้มข้อมูล

สำหรับข้อมูลที่จะนำมาวิเคราะห์นั้น นอกจากความถี่ของการพบคำในเอกสารแล้ว ตำแหน่งของคำที่ปรากฏในเอกสารก็เป็นสิ่งที่เราให้ความสนใจเช่นเดียวกัน

จากสมมติฐานที่ว่า คำที่พบในส่วนต้นของเอกสาร เช่น ส่วน Abstract, Title และส่วนท้ายของเอกสาร เช่น ส่วน Conclusion, Reference น่าจะเป็นคำที่เหมาะที่จะเป็น Index term ของเอกสารมากกว่าคำที่พบในส่วนกลางของเอกสาร ดังนั้นเราจึงมีการแบ่งข้อมูลที่จะเข้าสู่ Neural Network ออกเป็น 4 ส่วนตามรูปที่ 7-4 ได้แก่

- ความถี่ของการพบคำในส่วนต้นของเอกสาร (n < 1/5 ของจำนวนคำทั้งหมดในเอกสาร)
- ความถี่ของการพบคำในส่วนกลางของเอกสาร
  (n >= 1/5 ของจำนวนคำทั้งหมดในเอกสาร AND n <= 4/5 ของจำนวนคำทั้งหมดในเอกสาร)
- ความถี่ของการพบคำในส่วนปลายของเอกสาร (n > 4/5 ของจำนวนคำทั้งหมดในเอกสาร)
- จำนวนของคำทั้งหมดในเอกสาร



***รูปที่7-4 แสดงการแบ่งเอกสารเป็นส่วน เพื่อใช้ในการเลือก** Index terms*

ในส่วนนี้ต้องมีการทำ Text Operations ก่อน โดยทำส่วนแบ่งคำ, ส่วนกำจัด Stopwords, ส่วนกำจัด Suffix (Porter's Algorithm) ก่อน และคำที่กำจัด Suffix แล้วเหมือนกันจะรวมเข้าด้วยกัน ซึ่งในขั้นตอนนี้เราใช้การสร้าง Hash function โดยใช้อักษรตัวแรก ร่วมกับ การสร้าง Binary Tree รูปที่ 7-5 เพื่อใส่ข้อมูล ซึ่งจะช่วยลดระยะเวลาในการเปรียบเทียบระหว่างคำลงได้

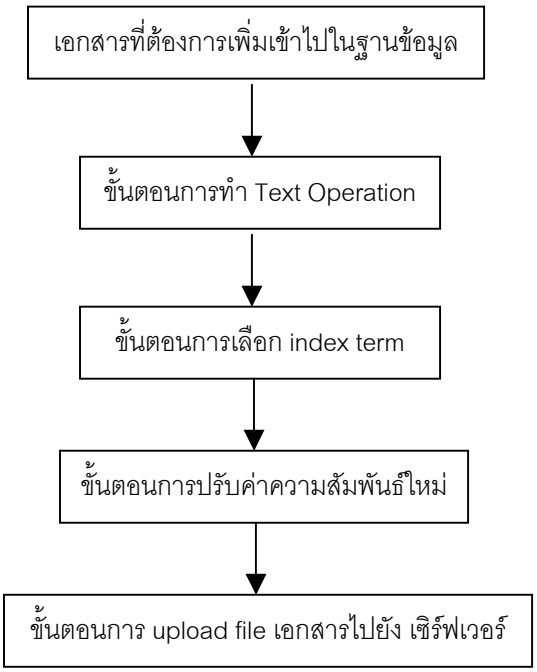
7.4 ส่วนการแก้ไขข้อมูลที่เกิดจากบรรณารักษ์

ส่วนนี้เป็นงานที่เกิดจาก บรรณารักษ์ หรือ ผู้ดูแลระบบซึ่งแบ่งเป็นงานย่อย ๆ ได้ออกเป็น 2 ส่วนด้วยกันคือ

1. งานในการเพิ่มเอกสารเข้าไปในระบบค้นคืนข้อมูล
2. งานในการแก้ไขข้อมูลของเอกสารซึ่งเก็บในฐานข้อมูล

7.4.1 งานในการเพิ่มเอกสารเข้าไปในระบบค้นคืนข้อมูล

โดยรูปแบบการ ใส่ข้อมูลเพิ่มไปยัง ฐานข้อมูลที่จัดเก็บไว้มีรูปแบบดังนี้



รูปที่ 7-6 แสดงขั้นตอนการเพิ่มเอกสารเข้าไปในระบบค้นคืนข้อมูล

จากรูปที่ 7-6 สามารถอธิบายรายละเอียดของขั้นตอนการเพิ่มเอกสารเข้าไปในระบบค้นคืนข้อมูลได้ดังนี้

<u>ขั้นตอนที่ 1</u> เอกสารที่ต้องการเพิ่มเข้าไปในฐานข้อมูล ขั้นตอนนี้เป็นขั้นตอนที่ บรรณารักษ์ต้องกรองข้อมูลบางส่วนเอง เช่น ชื่อเรื่อง คำอธิบายเกี่ยวกับ เอกสารเรื่องนั้น ผู้แต่งเอกสารนั้น โดยข้อมูลเหล่านี้จะถูกจัดเก็บลง ฐานข้อมูลเพื่อ ใช้ในการค้นคืน โดยการค้นคืนประเภทนี้จะไม่ได้ใช้ การค้นคืนแบบ ฟัซซี่ลอจิก แต่จะเป็นการค้นคืนในรูปแบบ extract เช่นผู้ใช้ อาจมี ความต้องการ ชื่อเรื่องที่มี คำว่า JSP programming เป็นส่วนประกอบของ ชื่อเรื่อง ในการหาเอกสารที่ผู้ใช้ป้อนเฉพาะ ชื่อเรื่องนั้น จะใช้ความสามารถของ ภาษา เอสคิวแอล(SQL) ในการค้นหา โดยในตัวอย่างนี้ จะใช้คำสั่ง เอสคิวแอล(SQL) คือ

```
Select * from Paper where description like  '%"jsp programming"%'
```

ประโยคภาษา เอสคิวแอล(SQL)ข้างบนนี้จะทำการค้นหาเอกสารที่ มีคำว่า “jsp programming” เป็นส่วนประกอบภายใน column description เช่นเดียวกับการค้นหาโดยใช้ชื่อผู้แต่งในการค้นหาจะทำในทำนองนี้เช่นกัน

<u>ขั้นตอนที่ 2</u> ขั้นตอนการทำ Text Operation เป็นขั้นตอนการทำให้ คำให้อยู่ในรูปแบบที่จะใช้ในการจัดเก็บโดยรายละเอียดในการทำ Text Operation สามารถดูได้ที่ บทที่ 4 จะเป็นขั้นตอนการทำ Text Operation และเมื่อทำให้คำ อยู่ในรูปแบบที่สามารถใช้ในการค้นหาได้แล้วนั้น จะส่ง คำ เหล่านี้ไปยังขั้นการทำ index อัตโนมัติ

<u>ขั้นตอนที่ 3</u> ขั้นตอนการทำ index อัตโนมัติ โดยในขั้นตอนนี้สามารถดูได้ที่ บทที่ 5 เรื่องการทำ index อัตโนมัติ และ หัวข้อ 7.3 ประกอบ และเมื่อเลือกค่าindex termได้แล้วนั้นindex termพวกนี้จะถูกไปจัดเก็บในฐานข้อมูล และ นำไปปรับค่าความสัมพันธ์ในขั้นตอนที่ 4

<u>ขั้นตอนที่ 4</u> ขั้นตอนการปรับค่าความสัมพันธ์ของindex term เมื่อได้รับค่า index termมาแล้วจะต้องนำไปปรับค่าในตารางความสัมพันธ์ของ index termที่อยู่ในตารางความสัมพันธ์

Index 1	Index 2	Relate
Cat	kitty	0.5
Cat	dog	0.25
Dog	puppy	0.5

โดยในการคำนวณนั้นถ้าเราคำนวณค่าความสัมพันธ์ใหม่ทั้งหมดจะทำให้เสียเวลานานมาก เพราะ การคำนวณหาความสัมพันธ์ในแต่ละครั้งนั้น ถ้ามีจำนวนindex term ทั้งหมด n จำนวน จะมีความซับซ้อนถึง O(n^2) ดังนั้นในการคำนวณแต่ละครั้งนั้นเราจะคำนวณเฉพาะ index termที่มีการเปลี่ยนแปลงเท่านั้น โดยหลักการทำก็ง่าย ๆ แต่ให้ความถูกต้องเท่ากันคือ จะเลือกindex termที่เกิดจากการเพิ่มเอกสารเข้าไปใหม่ ยกตัวอย่างเช่นมีindex termที่เกิดจากเอกสารใหม่ k เทอม ค่าความซับซ่อน จะคิดได้เป็น O(k^2) ซึ่งเวลาในการคำนวณจะแตกต่างกันมาก

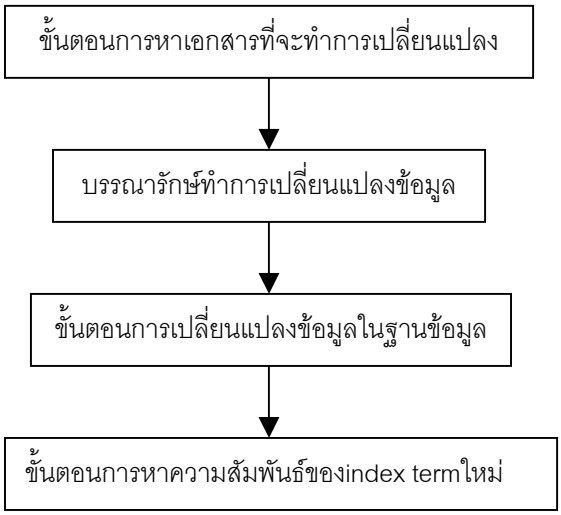
<u>ขั้นตอนที่ 5</u> ขั้นตอนการส่ง File เอกสารไปยัง เซิร์ฟเวอร์ ในขั้นตอนนี้จะทำการส่ง File ไปยัง เซิร์ฟเวอร์ โดยมีรูปแบบอินพุด FORM ดังนี้

```
<FORM METHOD="POST" ACTION="/jspsmartupload/jsp/sample1.jsp" ENCTYPE="multipart/form-data">
  <INPUT TYPE="FILE" NAME="FILE1" SIZE="50"><BR>
</FORM>
```

เมื่อทำการส่ง File เอกสารเสร็จสิ้นก็เป็นการเสร็จขั้นตอนการเพิ่มเอกสารเข้าไปในระบบค้นคืนข้อมูล

7.4.2 งานในการแก้ไขข้อมูลของเอกสารซึ่งเก็บในฐานข้อมูล

ข้อมูลของเอกสารในบางโอกาสจำเป็นต้องมีการแก้ไข เช่น แก้ไขชื่อผู้แต่ง แก้ไขชื่อเรื่อง หรือ แก้ไขindex term ซึ่งเป็นไปได้ทั้งสิ้นโดยเฉพาะการแก้ไขindex termนั้นเป็นสิ่งจำเป็นมาก เพราะ index termที่ได้จากการทำ อินเด็กซ์อัตโนมัติอาจไม่ถูกต้อง 100% ดังนั้นต้องมีกระบวนการที่เอื้ออำนวยความสะดวกให้แก่ บรรณารักษ์ หรือ ผู้ดูแลระบบ ในการเปลี่ยนแปลงข้อมูลที่จัดเก็บได้ง่าย ๆ โดยขั้นตอนในการแก้ไขข้อมูลของเอกสาร สามารถใช้รูปข้างล่างนี้ในการอธิบายได้



รูปที่ 7-7 ขั้นตอนงานในการแก้ไขข้อมูลของเอกสารซึ่งเก็บในฐานข้อมูล

จากรูปที่ 7-7 สามารถอธิบายรายละเอียดของขั้นตอนงานในการแก้ไขข้อมูลของเอกสาร ได้ดังนี้

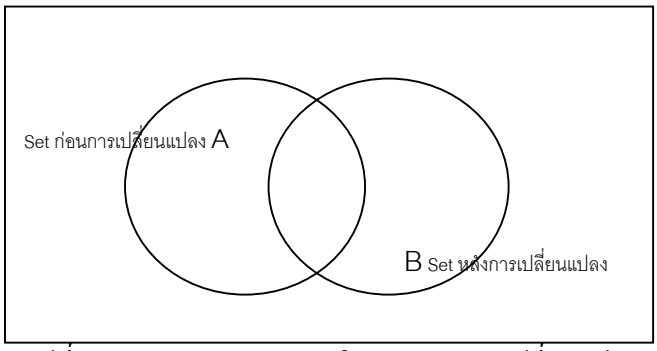
<u>ขั้นตอนที่ 1</u> ขั้นตอนการหาเอกสารเพื่อทำการเปลี่ยนแปลง ขั้นตอนนี้จะรับ อินพุด ของบรรณารักษ์ หรือ ผู้ดูแลระบบ โดยจะนำอินพุดที่ได้นั้นไปหารูปแบบของภาษา เอสคิวแอล(SQL) ที่ใช้ในการค้นการเอกสารได้โดยภาษา เอสคิวแอล(SQL)จะมีรูปแบบดังนี้

Select * from paper where “----------“

โดยส่วนที่ขีดเส้นนั้นหมายถึงจำนำ อินพุด ไปเขียนในส่วนของ where เพื่อใช้ในการเลือกข้อมูลในฐานข้อมูลออกมาแสดงในแก่ บรรณารักษ์ หรือ ผู้ดูแลระบบ แล้ว บรรณารักษ์ จะทำการเลือกเอกสารที่ต้องการแก้ไขอีกที เพราะ ค่า Recordset ที่ถูกเลือกออกมาอาจไม่ใช่เป็นค่าเดียวแต่อาจเป็น set ของ record ก็เป็นได้ เมื่อทำการเลือกเอกสารแล้วจะนำข้อมูลภายในการจัดเก็บของเอกสารฉบับ นั้น ๆ มาแสดง

<u>ขั้นตอนที่ 2</u> ขั้นตอนการเปลี่ยนแปลงข้อมูล จะนำตัวแปรมาจดจำ เซต ของ index ก่อนการเปลี่ยนแปลงไว้ก่อน หลังจากนั้นบรรณารักษ์จะทำการเปลี่ยนแปลงข้อมูลได้ทั้ง ชื่อผู้แต่ง ชื่อเรื่อง หรือ index term

<u>ขั้นตอนที่ 3</u> ขั้นตอนการเปลี่ยนแปลงข้อมูลในฐานข้อมูล หากข้อมูลผู้แต่ง หรือ ข้อมูล ชื่อเรื่อง เปลี่ยนแปลงจะทำได้โดยง่าย โดยการส่ง ภาษา เอสคิวแอล(SQL) ไป update ข้อมูลในฐานข้อมูลเท่านั้น แต่ การเปลี่ยน แปลงอินเด็กเทอมนั้น จะต้องมีขั้นตอนการคำนวณหาความสัมพันธ์ของindex termมาเกี่ยวข้อง ด้วยดังนั้นต้องมีโครงสร้างของโปรแกรมที่ช่วยในการคำนวณหาค่าความสัมพันธ์ใหม่เนื่องจากการเปลี่ยนแปลง index term โดย จะมีการเก็บค่า set ของindex term ก่อนและหลังการเปลี่ยนแปลง เพื่อช่วยให้การคำนวณหาความสัมพันธ์ของindex termได้มีความเร็วมากขึ้นโดยมีแนวคิดดังนี้ดังแสดงในรูปที่ 7-8



รูปที่ 7-8 เซตของindex termก่อนและหลังการปลี่ยนแปลง

โดยในการคำนวณนั้นจะคำนวณเฉพาะส่วนที่มีการเปลี่ยนแปลงเท่านั้นโดยจะคิดส่วนที่ลบออกก่อน ในที่นี้คือ A-B แล้วนำไปคำนวณหาค่าความสัมพันธ์ใหม่ และ คำนวณหาค่าความสัมพันธ์ของindex termใหม่ โดยใช้ส่วน B-A ในการคำนวณหาค่าความสัมพันธ์ใหม่

7.5 การออกแบบฐานข้อมูล

PAPER

←----p,k.----→

Pnumber	Pname	Abstract	IndexRef	Location	Filesize

ตาราง paper ทำหน้าที่เก็บข้อมูลของเอกสารแต่ละเอกสาร

- Pnumber เป็นคีย์หลักทำหน้าที่ในการอ้างอิงเอกสาร
- Pname ชื่อหัวเรื่องของเอกสาร
- Abstract บทคัดย่อของเอกสาร

- IndexRef index term อ้างอิงเพื่อใช้ในการแก้ไขเพราะ index ที่ใช้ในการหาค่าความสัมพันธ์จะอยู่ในรูปแบบที่เป็นรากศัพท์ ดังนั้นผู้แก้ไขจะไม่ทราบที่มาของ index term นั้น

AUTHOR

←---------- p,k. ----------→

←----------f,k----------→

Pnumber	Authorname

ตาราง Author ทำหน้าที่เก็บข้อมูลผู้แต่งเนื่องจากเอกสารแต่ละฉบับ อาจมีผู้แต่งได้มากกว่า 1 ท่าน

- Pnuber หมายเลขกำกับเอกสารเพื่อใช้ในการค้นหา
- Authorname ชื่อผู้แต่ง

PAPERKEY

←----------p,k----------→

←----------f.k----------→

Pnumber	Keyword

ตาราง PAPERKEY ใช้สำหรับเก็บ index term ของแต่ละเอกสารนั้น

- Pnuber หมายเลขกำกับเอกสารเพื่อใช้ในการค้นหา
- Keyword index term

RELATION

←---------- p,k. ----------→

Index1	Index2	Relate

ตาราง RELATION ใช้สำหรับเก็บคู่ความสัมพันธ์ของ index term

- Index1,Index2 คู่ Index term
- Relate ค่าความสัมพันธ์มีค่าอยู่ระหว่าง 0-1