บทที่ 2

แนวคิดและทฤษฎี

2.1 แนวคิดและทฤษฎี

ในอดีตการค้นคืนสารสนเทศจะดำเนินการด้วยคน ต่อมาได้มีการนำคอมพิวเตอร์มาประยุกต์ เพื่อใช้กับระบบค้นคืนสารสนเทศ ข้อดีของระบบค้นคืนสารสนเทศด้วยคอมพิวเตอร์ คือช่วยหลีกเลี่ยงความซับซ้อน และค่าใช้จ่ายในการสร้างดัชนีจากคน

ระบบค้นคืนสารสนเทศด้วยคอมพิวเตอร์ได้รับความสนใจในหลายแอปพลิเคชั่น เช่น ห้องสมุด สำนักงานอัตโนมัติ วิศวกรรมซอฟต์แวร์ พจนานุกรมคอมพิวเตอร์ เอนไซโคปีเดียคอมพิวเตอร์ และการเก็บเอกสารในสำนักงาน

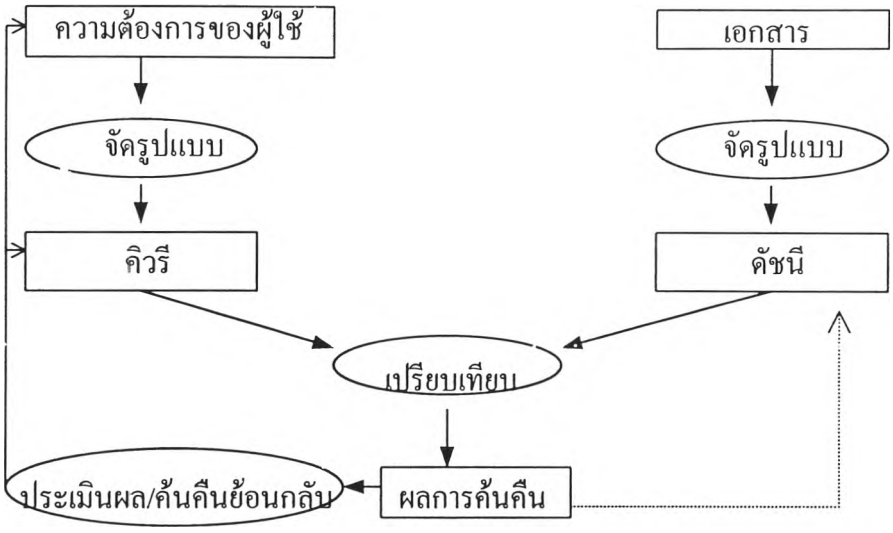
ผลลัพธ์จากการค้นคืนของระบบค้นคืนสารสนเทศ จะเป็นคำตอบที่พอใจได้นั้น มักไม่สามารถได้จากคิวรีง่ายๆ ผู้ใช้ต้องคิดด้วยตัวเอง เพื่อเดาว่า ข้อความใด น่าจะอยู่ในเอกสารที่ผู้ใช้ต้องการ ซึ่งเป็นจุดเสีย เพราะผู้ใช้ไม่ใช่ผู้เชี่ยวชาญ ในการสร้างดัชนี ด้านภาษาและไม่รู้ลักษณะของเอกสารที่ตนต้องการหา ทำให้ได้มีการคิดค้นวิธีเพื่อให้ได้ผลจากการค้นหาที่ทำให้ผู้ใช้พอใจที่สุดขึ้น เช่นการให้น้ำหนักคำ การจัดลำดับและการค้นคืนย้อนกลับ เป็นต้น

ระบบสืบค้นสารสนเทศในปัจจุบันมี 2 ระบบใหญ่ๆ คือ

1.ระบบค้นคืนแบบสัญนิยม (Conventional Retrieval System) เป็นระบบที่ใช้เทคโนโลยีพื้นฐานของแฟ้มข้อมูลผกผัน (Inverted files) ช่วยในการค้นคืนสารสนเทศ

2.ระบบค้นคืนชั้นสูง (Advanced Retrieval System) เป็นระบบที่ใช้เทคโนโลยีพื้นฐานของแบบจำลองปริภูมิเวกเตอร์ (Vector Space Model) ช่วยในการค้นคืนสารสนเทศ

องค์ประกอบของระบบค้นคืนข้อมูล ประกอบด้วยส่วนสำคัญดังรูปที่ 2.1



รูปที่ 2.1 แสดงองค์ประกอบของระบบค้นคืนข้อมูล

จากรูปที่ 2.1 เริ่มต้นด้วยการนำเอกสารที่ต้องการค้นคืนมาจัดรูปแบบเพื่อให้เหมาะสมกับการนำมาสร้างดัชนี แล้วจึงนำเอกสารมาสร้างเป็นดัชนีเพื่อใช้ค้นคืน ส่วนทางผู้ใช้ เมื่อมีความต้องการค้นคืนเอกสาร จะแสดงรูปแบบความต้องการเป็นรูปแบบของดิวรี การค้นคืนจึงนำดิวรีที่ได้จากผู้ใช้มาเปรียบเทียบกับดัชนี จะได้ผลการค้นคืนออกมา ให้ผู้ใช้ประเมินผลว่าเอกสารที่ค้นคืนมาได้ ตรงตามต้องการหรือไม่ ถ้าผลที่ได้ผู้ใช้ยังไม่พอใจ ผู้ใช้สามารถค้นคืนย้อนกลับอีกครั้ง เพื่อให้ได้ผลที่น่าพอใจขึ้น นอกจากนี้ในบางระบบผลการค้นคืนสามารถนำเอาไปปรับปรุงดัชนีได้ด้วย

2.2 โครงสร้างข้อมูลที่ใช้ในระบบค้นคืนสารสนเทศ

ให้เอกสารเป็นข้อมูลใน 1 แฟ้ม หรือหลายๆแฟ้ม และเมื่อเอกสารใหญ่มากๆ จำเป็นต้องมีการสร้างดัชนี เพื่อช่วยพัฒนาการค้นหาให้รวดเร็วขึ้น โดยการสร้างดัชนีจะสร้างบนเอกสารที่ถูกนอมัลไลซ์ ซึ่งก็คือเอกสารที่ถูกประมวลผลเสร็จแล้วจากเอกสารต้นฉบับ ออกมาในรูปแบบที่ผู้ใช้กำหนดขึ้น

ในที่นี้จะกล่าวถึงโครงสร้างที่ใช้เอกสารซึ่งตัวดัชนี จะถูกเพิ่มข้อมูลที่เกี่ยวกับเอกสารเข้าไป เพื่อให้สามารถค้นหาได้เร็วกว่าค้นจากเอกสารต้นฉบับโดยตรง ซึ่งแฟ้มข้อมูลที่ถูกเพิ่มดัชนีนั้นมีโครงสร้างที่สำคัญดังนี้

1.แฟ้มผกผัน (Inverted file) เป็นโครงสร้างข้อมูลที่อาศัยตารางเป็นกลไกในการเข้าถึงข้อมูล โดยตารางดังกล่าวประกอบด้วยคำหลักและตัวชี้ ที่ชี้ไปยังข้อมูลในแฟ้มข้อมูลตามคำหลัก

นั้นๆ เป็นโครงสร้างที่มีรูปแบบง่าย ไม่ซับซ้อน ทำให้ง่ายในการเขียนโปรแกรม แต่มักจะใช้เนื้อที่จัดเก็บสูง เพราะต้องนำส่วนของข้อมูลที่เป็นดัชนีไปเก็บไว้ในแฟ้มผกผันด้วย

2.แฟ้มซิกเนเจอร์ (Signature file) เป็นการค้นหาแบบแฮช (hashing) โดยแปลงข้อความในเอกสารเป็นลำดับบิต จะได้ข้อความที่ถูกบีบอัด (10%-20% ของเอกสารต้นฉบับ) ส่วนคิวรีจะถูกแปลงเป็นลำดับบิต เพื่อหาในดัชนี ทำให้มีการค้นคืนได้รวดเร็วเพราะอาศัยการคำนวณฟังก์ชั่นแฮช แต่เพราะว่ามีโอกาสที่มีคำตอบมากกว่า 1 ค่าที่ชี้ที่เดียวกัน ทำให้คำตอบที่ได้ไม่ถูกต้อง ถึงแม้จะเลือกสูตรที่เหมาะสมก็ทำได้ในเอกสารขนาดกลางๆเท่านั้น วิธีนี้ไม่เหมาะกับเอกสารขนาดใหญ่ๆ หรือ เอกสารที่ไม่สามารถแบ่งคำได้ เช่นเอกสารที่มีข้อความเป็นลำดับโปรตีน(DNA) เป็นต้น

3.ต้นไม้แพ็ต (Pat tree) คือต้นไม้แพ็ตทิเชีย (Patricia) ย่อมาจาก Practical Algorithm to Retrieve Information Coded In Alphanumeric (Knuth , 1973; Sedgewick, 1988; Wood, 1993) เป็นต้นไม้ดิจิตอล แบบทวิภาค (Binary Digital Tree) หรือ ทรัยแบบทวิภาค (Binary tries) ต้นไม้แพ็ต เป็นโครงสร้างข้อมูลที่มีประสิทธิภาพมากกับการค้นข้อมูลที่มีการสร้างดัชนีเตรียมไว้ก่อน (Preprocessing) เพื่อการค้นข้อมูลในภายหลัง (Gonnet and Baeza, 1991) และในปี 1985 ศูนย์คอมพิวเตอร์ของ Oxford English Dictionary (OED)ได้พัฒนาระบบแพ็ต (PAT System) ขึ้นจนในเวลาต่อมาได้รู้จักกันอย่างแพร่หลายว่าเป็นระบบที่สามารถค้นหาข้อมูลได้รวดเร็วมาก

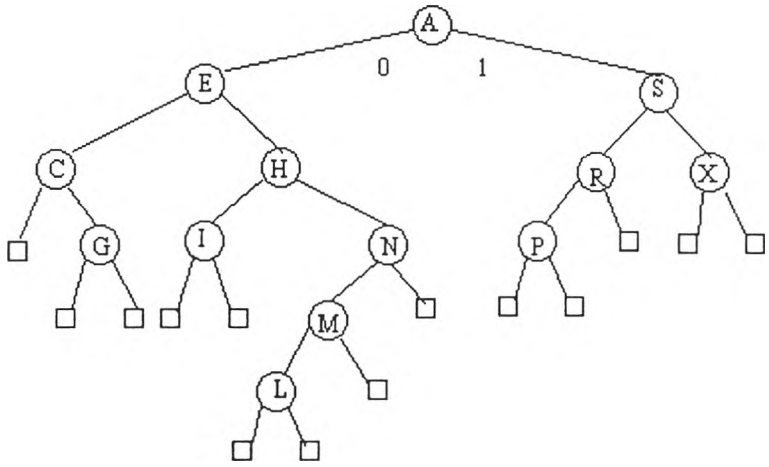
2.3 โครงสร้างต้นไม้แพ็ต (PAT tree)

โครงสร้างพื้นฐานของต้นไม้แพ็ตเป็นต้นไม้แบบดิจิตอล ซึ่งเป็นโครงสร้างแบบต้นไม้ที่เส้นทางเดิน ระหว่างโหนดของต้นไม้ เป็นการแทนค่าตัวอักษรแต่ละตัวที่ประกอบเป็นคำศัพท์ โดยต้นไม้แบบดิจิตอลมีความแตกต่างกับต้นไม้ที่ไม่ใช่ดิจิตอล (เช่นต้นไม้แบบทวิภาค) ตรงที่เส้นทางระหว่างโหนดของต้นไม้ดิจิตอลอาศัยผลลัพธ์จากการคำนวณระหว่างตัวเลข ไม่ได้ตรวจสอบจากการเปรียบเทียบค่าดัชนีระหว่างโหนดกับค่าที่ต้องการค้นหาอย่างเช่นในต้นไม้ที่ไม่ใช่ดิจิตอล ซึ่งมีโครงสร้างดังรูปที่ 2.2

กำหนดให้	A	00001	N	01110
	S	10011	G	00111
	E	00101	X	11000
	R	10010	M	01101
	C	00011	P	10000

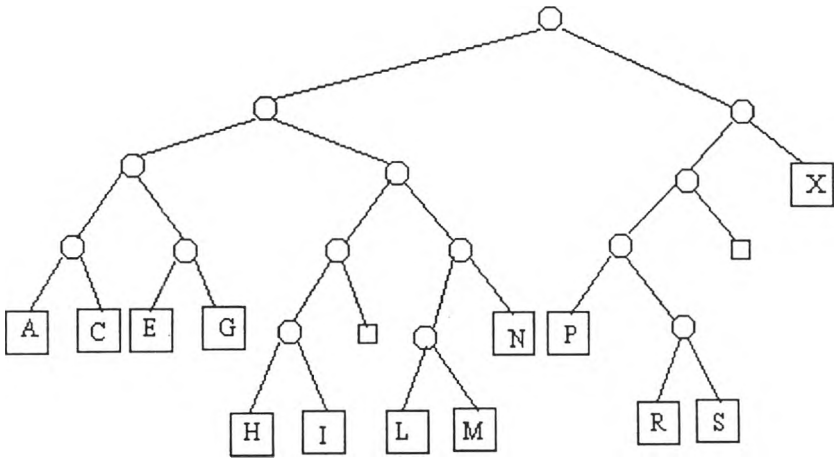
H 01000 L 01100

I 01001



รูปที่ 2.2 แสดงต้นไม้ดิจิตอล

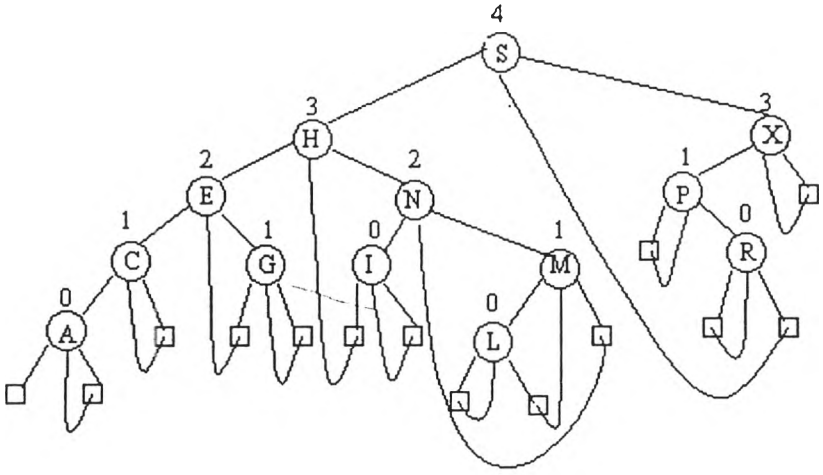
รูปที่ 2.2 เป็นตัวอย่างต้นไม้ดิจิตอลที่สร้างขึ้นตามลำดับข้อมูลคือ A, S, E, R, C, H, I, N, G, X, M, P, L ตามลำดับ โดยการค้นหาข้อมูลในต้นไม้ยังคงอาศัยการเปรียบเทียบค่าในแต่ละโหนดที่เดินทางผ่าน ซึ่งต้นไม้แบบดิจิตอลสามารถลดขั้นตอนการเปรียบเทียบ โดยการเก็บดัชนีไว้ที่โหนดภายนอก (External node) เท่านั้น ซึ่งวิธีนี้เรียกว่า ทรัย (Trie) ซึ่งมีโครงสร้างดังรูปที่ 2.3



รูปที่ 2.3 แสดงทรัย

รูปที่ 2.3 จะเห็นว่าค่าดัชนีถูกเก็บอยู่ที่โหนดสี่เหลี่ยม เรียกว่าโหนดภายนอก ส่วนโหนดวงกลมไม่เก็บค่าดัชนี เรียกว่าโหนดภายใน ดังนั้นในการค้นหาข้อมูลจะต้องค้นจากโหนดรากถึงโหนดภายนอกเสมอ โดยมีการเปรียบเทียบที่โหนดภายนอกเพียงครั้งเดียวต่อการค้นหาหนึ่งครั้ง ซึ่งเร็วกว่าการค้นหาของต้นไม้ดิจิตอลรูปที่ 2.2 แต่ทรัยมีข้อเสียคือใช้เนื้อที่จัดเก็บต้นไม้มากกว่า

โครงสร้างต้นไม้แพ็ตจะนำข้อดีของทรัยที่ลดขั้นตอนการเปรียบเทียบทุกๆโหนด โดยเก็บค่าข้อมูลไว้ที่โหนดภายนอกแทน รวมกับการประหยัดเนื้อที่จัดเก็บต้นไม้ โดยนำหลักการบิตข้าม (Skip bit) มาใช้ทำให้ลดขนาดต้นไม้ได้ ซึ่งมีโครงสร้างดังรูปที่ 2.4



รูปที่ 2.4 แสดงต้นไม้แพ็ต

รูปที่ 2.4 จะเห็นว่าต้นไม้แพ็ตมีบิตข้ามเป็นตัวระบุว่าจะตรวจสอบบิตใดของข้อมูล เพื่อเดินทางในต้นไม้อย่างถูกต้อง โดยบางครั้งอาจข้ามบิตได้ทำให้ค้นหาเร็วขึ้น โดยมีการเปรียบเทียบที่โหนดภายนอกเพียงครั้งเดียว ซึ่งต้นไม้แพ็ตมีคุณสมบัติดังนี้คือ

1 ต้นไม้ย่อยไม่มีโหนดว่าง (null node)

2 ถ้าโหนดภายในเป็น 0 จะแยก (branch) ไปทางต้นไม้ย่อยข้างซ้าย แต่ถ้าโหนดภายในเป็น 1 จะแยกไปทางต้นไม้ย่อยข้างขวา

3 โหนดภายนอกเก็บค่าตัวชี้ที่ชี้ข้อมูล

2.4 สายอักขระแบบเซมิอินไฟไนต์ (Semi-infinite string) หรือ ซิสตริง (Sistring)

โดยทั่วๆ ไปแล้วเรามองข้อมูลเป็น ลำดับสายอักขระ (Array of character) ที่เรียงต่อกันไปเรื่อยๆ จนจบข้อมูล สำหรับ ซิสตริง ก็คือ ลำดับย่อย (Subsequence) ต่างๆ ในลำดับสายอักขระ โดยเริ่มที่จุดเริ่มต้น แล้วเลื่อนไปทางขวาจนจบข้อมูล ซึ่งคำว่า ซิสตริง มาจากคำว่า Semi-infinite line โดยคำว่าบรรทัด (Line) หมายถึง จุดเริ่มต้นหนึ่งแล้วเลื่อนไปทางทิศใดทิศหนึ่งไม่มีที่สิ้นสุด จะเห็นว่า ซิสตริงในข้อมูลหนึ่งๆ จะมีตัวเดียว (Unique) เสมอ ดังตัวอย่างต่อไปนี้

Text	Once upon a time, in a far away land…
Sistring 1	Once upon a time, in a far away land…
Sistring 2	nce upon a time, in a far away land…
Sistring 8	on a time, in a far away land…
Sistring 11	a time, in a far away land…
Sistring 22	a far away land…

และในการเปรียบเทียบซิสตริงสองตัว จะเปรียบเทียบตามลำดับรหัสค่าของซิสตริงนั้น เช่น ลำดับแอสกี (ASCII ordering) จะได้ผลลัพธ์ตามตัวอย่างข้างบนดังนี้

Sistring22 < Sistring11 < Sistring2 < Sistring8 < Sistring1

โดยมี ซิสตริง " a far away land…" เป็นค่าน้อยสุด ส่วนซิสตริง "upon a time, in a far away land…" เป็นค่ามากที่สุด

2.5 โครงสร้างแถวลำดับแพ็ต (PAT array)

ต้นไม้แพ็ตมีจุดเด่น คือ ไม่อาศัยคำหลัก (Keyword) จึงไม่มีการเลือกคำเกิดขึ้นในระบบ และสามารถจัดการกับข้อมูลที่มีข้อมูลร่วม (Prefix) ซ้ำๆ กันได้ดี โดยเก็บรวมกันไว้ที่เดียว จึงทำให้ลดขนาดของการจัดเก็บได้มาก อีกทั้งยังทำให้การค้นคืนข้อมูลมีประสิทธิภาพสูงอีกด้วย แต่เนื่องจากโครงสร้างของต้นไม้แพ็ต แต่ละโหนดจำเป็นต้องเก็บ ตัวชี้ (Pointer) เพื่อใช้ในการแยก (Branch) ไปยังโหนดต่อไป ทำให้ใช้เนื้อที่ในการจัดเก็บมาก แต่เราสามารถลดปริมาณเนื้อที่เหล่านี้ลง โดยอาศัยแถวลำดับเข้ามาช่วยได้ ซึ่งเราเรียกโครงสร้างข้อมูลนี้ว่า แถวลำดับแพ็ต ซึ่งผู้คิดค้นวิธี

การนี้คือ แมนเบอร์ และไมเยอร์ (Manber and Myers , 1990) และได้ตั้งชื่อว่า ซัฟฟิคอะเรย์ (Suffix array) หรือแถวลำดับแพ็ต โดยมีหลักการ คือนำโหนดภายนอกของต้นไม้แพ็ตทิเชีย (Patricia) มาเก็บลงในแถวลำดับเดี่ยว (Single array) เรียงต่อๆ กันไป ผลที่ได้คือ ซิสตริง ต่างๆถูกเก็บเรียงต่อกันไปตามลำดับอยู่ในแถวลำดับ โดยการค้นหาบนแถวลำดับแพ็ต อาศัยการค้นแบบทวิภาคกับแถวลำดับ จะมีประสิทธิภาพประมาณ O(log n) โดยที่ n คือจำนวนโหนดทั้งหมดในต้นไม้

2.6 การวัดประสิทธิภาพของระบบค้นคืนสารสนเทศ

ในการค้นคืนสารสนเทศจะมีการวัดปริมาณที่สำคัญ 2 ค่าคือค่าความถูกต้อง (Precision) และ ค่าเรียกคืน (Recall) โดยที่ทั้งสองค่า มีสูตรการคำนวณ ดังนี้คือ

$$\text{ค่าความถูกต้อง} = \frac{\text{จำนวนเอกสารตรงตามต้องการที่ค้นคืนออกมาได้}}{\text{จำนวนเอกสารทั้งหมดที่ค้นคืนออกมา}}$$

$$\text{ค่าเรียกคืน} = \frac{\text{จำนวนเอกสารตรงตามต้องการที่ค้นคืนออกมาได้}}{\text{จำนวนเอกสารตรงตามต้องการทั้งสิ้นในฐานข้อมูล}}$$

ค่าความถูกต้องเป็นปริมาณที่แสดงว่าการค้นคืนเอกสารได้ตรงตามต้องการเพียงใดเช่น ถ้าค้นคืนเอกสารออกมาได้ N ฉบับ และมีเอกสารอยู่ R ฉบับที่ตรงตามต้องการ ดังนั้นค่าความถูกต้องมีค่าเป็น R/N หรือเป็นโอกาสของเอกสารที่ค้นคืนออกมาตรงตามต้องการ ส่วน ค่าเรียกคืน เป็นปริมาณที่แสดงความครอบคลุม (Thoroughness) เช่น ถ้าฐานข้อมูลมีเอกสารที่ตรงตามต้องการทั้งสิ้น T ฉบับ และการค้นคืนสามารถดึงเอกสารที่ตรงตามต้องการได้ R ฉบับ ค่าเรียกคืนเป็น R/T ทั้งค่าเรียกคืนและค่าความถูกต้อง มีค่าอยู่ระหว่าง 0 ถึง 1ในทางอุดมคติ การค้นคืนต้องการให้ได้เฉพาะเอกสารที่ตรงตามต้องการเท่านั้น ซึ่งในกรณีนี้ค่าของทั้ง ค่าเรียกคืน และ ค่าความถูกต้อง มีค่าเป็น 1 ซึ่งในทางปฏิบัติเป็นไปได้ยาก จากผลการวิจัย (Salton and McGill, 1983) พบว่าค่าเรียกคืน และ ค่าความถูกต้อง มีความสัมพันธ์กันแบบปฏิภาคผกผัน คือหากต้องการให้ค่าเรียกคืนสูง ค่าความถูกต้อง ก็จะต่ำ และในทางตรงข้าม หากต้องการให้ค่าความถูกต้องสูง ค่าเรียกคืนก็จะต่ำ

ตัวอย่างผลค่าเรียกคืนและค่าความถูกต้อง (Salton and McGill, 1983) จากการค้นคืน ซึ่งเป็นผลลัพธ์การค้นคืนของคิวรีหนึ่ง โดยได้พบเอกสารที่มีคำที่ตรงกับคิวรีจำนวน 14 เอกสาร แต่มีเอกสารที่ตรงตามต้องการเพียง 5 เอกสาร โดยสามารถคำนวณค่าเรียกคืน และค่าความถูกต้อง ได้ดังตารางที่ 2.1

ลำดับ	หมายเลขเอกสาร	ค่าเรียกคืน	ค่าความถูกต้อง
1	588 *	0.2	1.0
2	589 *	0.4	1.0
3	687	0.4	0.67
4	456 *	0.6	0.75
5	987	0.6	0.60
6	342 *	0.8	0.67
7	356	0.8	0.57
8	455	0.8	0.50
9	466	0.8	0.44
10	788	0.8	0.40
11	987	0.8	0.36
12	342	0.8	0.33
13	233 *	1.0	0.38
14	566	1.0	0.36

* แสดงว่าเอกสารนั้นตรงตามต้องการ

ตารางที่ 2.1 แสดงผลการค้นคืน

เนื่องจากผลการใช้ค่าความถูกต้องและค่าเรียกคืน ไม่สามารถเปรียบเทียบระบบได้ชัดเจน จึงได้มีการวัดประสิทธิภาพระบบโดยใช้เพียงค่าเดียว คือค่าเฉลี่ยความถูกต้องหรือค่าเฉลี่ยเรียกคืน (โดยทั่วไปจะเลือกใช้ค่าเฉลี่ยความถูกต้องมากกว่า) ซึ่งค่าเฉลี่ยความถูกต้องได้มาจากการเฉลี่ยค่าความถูกต้องที่ค่าเรียกคืน ลำดับต่างๆกัน ตั้งแต่ 0.1, 0.2, … , 1.0 จำนวน 10 ลำดับ

2.7 ความสำคัญของคำ (Term importance)

เมื่อเกิดการค้นหาคำพบในเอกสาร คำเหล่านั้นจะถูกให้ค่าความสำคัญต่างๆกัน เช่นในภาษาอังกฤษ คำว่า “the” มีจำนวนมากในเอกสาร แต่ก็ไม่ได้เป็นการแสดงว่าคำนั้นเป็นประโยชน์

ต่อผู้ใช้ ในทางตรงกันข้าม ถ้าเป็นคำว่า "telecommunications" ถูกค้นพบ คำนี้จะให้ประโยชน์ต่อผู้ใช้มากกว่า การพบคำว่า "the" สิ่งเหล่านี้แสดงว่า คำที่แตกต่างกันควรมีค่าความสำคัญต่างกันในการค้นคืนด้วย

2.8 การให้น้ำหนักคำ (Term weighting)

การให้น้ำหนักคำเป็นวิธีการประมาณค่าความสำคัญของคำ การให้น้ำหนักคำที่เหมาะสมสามารถเพิ่มประสิทธิภาพของระบบค้นคืนข้อมูลได้ โดยนำค่าที่ได้ไปใช้เรียงลำดับเอกสาร เพื่อบอกถึงลำดับประโยชน์ของเอกสารเหล่านั้น ตามความต้องการของผู้ใช้

องค์ประกอบของการให้ความสำคัญของคำในระบบค้นคืนข้อมูล มี 3 ส่วนคือ

1 ความถี่ของคำ (Term frequency: tf) เอกสารที่มีคำที่ผู้ใช้ต้องการมากกว่า จะเป็นเอกสารที่มีประโยชน์ต่อผู้ใช้สูงกว่าด้วย เช่นต้องการหาคำว่า "telecommunications" เอกสารที่ใช้คำว่า "telecommunications" 10 ครั้ง จะเป็นประโยชน์มากกว่าเอกสารที่มีคำว่า "telecommunications" เพียงครั้งเดียว

2 ความถี่ของเอกสารแบบผกผัน (Inverse Document Frequency: IDF) เนื่องจากคำทั่วๆไป (เช่นคำที่ใช้มากๆ ในแต่ละเอกสาร) จะมีความสำคัญน้อยกว่าคำที่ไม่ทั่วไป (เช่นคำที่ใช้เฉพาะบางเอกสาร) เช่น การพบคำว่า "the" เป็นประโยชน์ต่อผู้ใช้น้อยกว่าการพบคำว่า "telecommunications"

3 ความยาวของเอกสาร (Document length) เอกสารที่ยาวจะมีจำนวนคำมาก และความถี่ของคำสูง ดังนั้นเอกสารที่ยาวจึงมีโอกาสถูกค้นคืนมากกว่าเอกสารที่สั้น เพราะความยาวเอกสารต่างกัน ไม่ใช่เพราะประโยชน์ต่อผู้ใช้ของเอกสาร เพื่อขจัดข้อได้เปรียบของเอกสารที่มีขนาดยาว จึงมีการทำนอร์มัลไลเซชัน (Normalization) ความยาวเอกสาร เพื่อชดเชยความยาวของเอกสารที่มีความยาวแตกต่างกัน

สูตรการให้น้ำหนักคำจำนวนมาก นำทั้ง 3 องค์ประกอบมาใช้หาค่าความสำคัญของคำ สำหรับรายละเอียดขององค์ประกอบทั้ง 3 มีดังนี้

1 ความถี่ของคำ

ความสำคัญของคำ เพิ่มตามจำนวนคำที่พบในเอกสาร ฉะนั้นเป็นฟังก์ชันแปรผันตรงต่อกัน ระหว่างน้ำหนักของคำ กับจำนวนคำที่พบ โดยจำนวนคำที่พบในเอกสารนั้น เราเรียกว่า "ความถี่ของคำ" ฟังก์ชันความถี่ของคำมีดังนี้

1.1 ความถี่ของคำแท้จริง (raw term frequency) โดยใช้จำนวนครั้งที่ปรากฏคำนั้นในเอกสาร

1.2 ความถี่ของคำแบบล็อกการิทึม (logarithmic term frequency) วิธีนี้ใช้ความถี่ของคำในฟังก์ชันของล็อกการิทึม ซึ่งมีรูปแบบเป็น

$$tf = 1 + \ln(tf)$$

โดย tf เป็นความถี่ของคำ

จากความจริงที่ว่าถ้าเอกสารหนึ่งพบคำตามคิวรีเพียงหนึ่งคำ แต่มีค่าความถี่ของคำสูง ต้องไม่สำคัญกว่า เอกสารอีกอันหนึ่งซึ่งพบคำตามคิวรีถึงสองตัว แต่มีค่าความถี่ของคำต่ำกว่า ตัวอย่างเช่น คิวรี "recycling of tires" กับสองเอกสาร D1 และ D2 โดยที่ D1 มีคำว่า "recycling" 10 ครั้ง และ D2 มีคำว่า "recycling" และ "tires" อย่างละ 3 ครั้ง ถ้าใช้ค่าความถี่แท้จริง D1 จะสำคัญกว่า (ค่าน้ำหนักคำเท่ากับ 10) D2 (ค่าน้ำหนักคำเท่ากับ 3+3 = 6) ซึ่งจะเห็นว่าโดยปกติ D2 ควรสำคัญกว่า D1 แต่ถ้าใช้ค่าแบบล็อกการิทึม D1 มีค่าความสำคัญเป็น 1+ln(10) = 3.3 ขณะที่ D2 มีค่าความสำคัญเป็น [1+ln(3)] + [1+ln(3)] = 4.1 ซึ่ง D2 จะมีค่าความสำคัญสูงกว่า D1

1.3 ความถี่ของคำแบบไบนารี (binary term frequency) วิธีนี้ไม่สนใจค่าความถี่ของคำ โดยมีค่าเพียงแค่ 1 หรือ 0 เท่านั้น โดยถ้ามีคำในเอกสารจะมีค่าเป็น 1 หรือไม่มีคำในเอกสารจะมีค่าเป็น 0

1.4 ความถี่ของคำแบบขยาย (augment term frequency) วิธีนี้ช่วยลดช่วงของค่าความถี่ของคำ โดยบีบค่าให้อยู่ระหว่างค่า 0.5 ถึง 1.0 โดยถ้าพบคำจะมีค่าเริ่มต้นที่ 0.5 แล้วเพิ่มค่าตามจำนวนคำที่พบ โดยค่ามากที่สุดเป็น 1.0 ค่าความถี่เป็นดังนี้

$$tf = 0.5 + 0.5 * tf / max\_tf$$

โดย max_tf เป็นความถี่ของคำสูงสุดในเอกสาร

2 ความถี่ของเอกสารแบบผกผัน

การใช้เพียงค่าความถี่ของคำมาประมาณค่าความสำคัญของคำนั้นยังไม่พอเพียง เช่นคำว่า "the" มีความถี่ของคำสูงมากในเอกสารจำนวนมาก แต่คำเหล่านี้กลับไม่มีความสำคัญมาก ดังนั้นต้องมีการแยกความสำคัญของคำเหล่านั้นออกมา

จะสังเกตเห็นว่า คำที่ถูกใช้ในเอกสารส่วนมาก จะเป็นคำทั่วๆไป และเป็นประโยชน์น้อยกว่า คำที่ถูกใช้ในเฉพาะเอกสารส่วนน้อย ซึ่งเป็นฟังก์ชันแปรผกผันของจำนวนเอกสารที่พบคำ โดยที่จำนวนเอกสารที่พบคำ เราเรียกว่าค่า "ความถี่ของเอกสาร" (Document frequency :df) ของคำ และฟังก์ชันผกผันของความถี่ของเอกสารจะเรียกว่า "ความถี่ของเอกสารแบบผกผัน"

สูตรที่ใช้หาค่าความถี่ของเอกสารแบบผกผันคือ

$$IDF = \log (N/df)$$

โดย N เป็นจำนวนเอกสารทั้งหมด
df เป็นความถี่ของเอกสารที่พบคำ

3 ความยาวของเอกสาร

แม้ว่าการประมาณค่าความสำคัญของคำในเอกสาร โดยใช้ความถี่ของคำและความถี่เอกสารแบบผกผันจะพอเพียงแล้วก็ตาม แต่ยังมีบางส่วนที่ถูกมองข้ามไป นั่นคือความยาวของเอกสาร ในความเป็นจริงแล้ว เอกสารมีขนาดความยาวต่างๆ กันไป และความยาวของเอกสารนี้เป็นข้อได้เปรียบกัน คือเอกสารยาวมีโอกาสที่ค่าความสำคัญของเอกสารสูงกว่าเอกสารสั้น ดังนั้นจึงต้องมีการชดเชยค่าความยาวเอกสารที่แตกต่างกัน ที่เรียกว่าการทำนอร์มัลไลเซชันความยาวเอกสาร เหตุผลที่ต้องมีการทำนอร์มัลไลเซชันความยาวของเอกสารคือ

- ความถี่ของคำสูงกว่า ปกติแล้วเอกสารที่ยาวจะมีการใช้คำซ้ำๆ กันมาก เป็นผลให้ค่าความถี่ของคำมีค่าสูงสำหรับเอกสารที่ยาว ซึ่งทำให้น้ำหนักคำในเอกสารยาวเพิ่มขึ้น
- จำนวนคำที่มากกว่า เอกสารยาวจะมีจำนวนคำที่แตกต่างกันเป็นจำนวนมาก ทำให้โอกาสพบคำในเอกสารยาวได้มากขึ้นกว่าในเอกสารสั้นๆ

จากปัญหาทั้งสองข้างต้น ฉะนั้นต้องมีการชดเชยค่าความยาวของเอกสารที่มีขนาดแตกต่างกัน การทำนอร์มัลไลเซชันความยาวของเอกสาร ทำเพื่อลดข้อได้เปรียบของเอกสารยาวที่มีโอกาสถูกค้นคืนสูงกว่าเอกสารสั้น ให้สามารถมีโอกาสในการถูกค้นคืนเท่าๆ กันได้ การทำนอร์มัลไลเซชันความยาวของเอกสาร เป็นวิธีการปรับน้ำหนักคำ ให้เหมาะสมกับความยาวของเอกสาร โดยมีวิธีการทำนอร์มัลไลเซชันดังนี้

3.1 การทำนอร์มัลไลเซชันแบบโคซายน์ (Cosine normalization) เป็นวิธีที่ไม่ได้ใช้ค่าความยาวเอกสาร แต่ใช้ค่าน้ำหนักของคำแทน ดังนี้

$$\text{Cosine normalization} = 1 / \left(\sqrt{\sum_{i=1}^{t} w_i^2} \right)$$

ให้ w_i เป็นค่าน้ำหนักคำโดย i มีค่าตั้งแต่ 1 ถึง t

t เป็นจำนวนคำทั้งหมดในเอกสาร

การทำนอร์มัลไลเซชันแบบโคซายน์ สามารถชดเชยได้ทั้งค่าความถี่คำและจำนวนคำ คือ ถ้า tf เพิ่ม ค่า w_i ก็เพิ่มด้วยทำให้แก้ปัญหาของความถี่ของคำ และถ้ามีจำนวนคำมากขึ้น จำนวนของ t ของคำก็จะเพิ่มขึ้นด้วย ทำให้แก้ปัญหาของจำนวนคำ

3.2 การทำนอร์มัลไลเซชันความถี่ของคำสูงสุด (Maximum term frequency normalization) เป็นวิธีซึ่งใช้ใน ระบบ สมาร์ท (SMART system) เป็นความถี่ของคำแบบขยาย ที่มีการใช้ความถี่ของคำสูงสุด แต่วิธีนี้ยังมีข้อจำกัดคือ วิธีนี้ช่วยชดเชยเฉพาะความถี่ของคำ แต่ไม่พิจารณาถึงโอกาสพบคำในเอกสารยาวที่มีจำนวนคำที่มากกว่า การทำนอร์มัลไลเซชันความถี่ของคำสูงสุด ทำได้โดยการหารน้ำหนักคำด้วยค่าความถี่ของคำสูงสุด ซึ่งคล้ายกับรูปแบบความถี่ของคำแบบขยาย

3.3 การทำนอร์มัลไลเซชันความยาวไบต์ (Byte length normalization) เป็นการทำนอร์มัลไลเซชันด้วยความยาวของเอกสารโดยที่ขนาดของเอกสารมีค่าเป็นไบต์ การทำนอร์มัลไลเซชันความยาวไบต์ สามารถชดเชยได้ทั้งค่าความถี่ของคำและจำนวนคำคือขนาดของไบต์ของเอกสารจะเพิ่มถ้าใช้จำนวนคำมากขึ้นหรือความถี่ของคำมากขึ้น การทำนอร์มัลไลเซชันความยาวไบต์ทำได้โดยการหารด้วยอัตราส่วนขนาดของเอกสารกับค่าเฉลี่ยขนาดของเอกสารทั้งหมด

จากองค์ประกอบทั้งสามของการให้น้ำหนักดังข้างต้นได้มีการวิจัยเกี่ยวกับน้ำหนักคำที่ศึกษามา ในที่นี้ผู้วิจัยเลือกสูตรน้ำหนักคำที่สำคัญมา 5 แบบดังนี้

1. แบบใช้ความถี่ของคำมาตราฐาน (Luhn ,1957) เป็นวิธีให้น้ำหนักคำแบบพื้นฐานโดยให้น้ำหนักสูงกับคำที่ปรากฏมากในเอกสาร และให้น้ำหนักคำลดลงสำหรับคำที่ปรากฏไม่มากในเอกสาร สูตรที่ใช้คือ

$$w_{ij} = tf_{ij}$$

โดย w_{ij} = ค่าน้ำหนักคำ i ในเอกสาร j
tf_{ij} = ค่าความถี่ของคำ i ในเอกสาร j

2. แบบใช้ค่าความถี่ของคำและจำนวนเอกสาร (Sparck Jones ,1972) เป็นวิธีที่นำจำนวนเอกสารมาช่วยปรับค่าน้ำหนักคำ ความสำคัญของคำจะลดถ้าจำนวนเอกสารที่คำปรากฏมากขึ้นโดยถ้าคำนั้นๆ ปรากฏในหลายๆ เอกสาร จะทำให้น้ำหนักคำลดลง สูตรที่ใช้คือ

$$w_{ij} = tf_{ij} / n_i$$

โดย n_i = จำนวนของเอกสารทั้งหมดที่มีคำศัพท์ i ในฐานข้อมูล

3. แบบใช้ค่าความถี่ของคำและความถี่เอกสารแบบผกผันหรือ IDF (Salton and Yang, 1973) เป็นสูตรค่าน้ำหนักคำที่นิยมใช้กันมากซึ่งเป็นรูปแบบของผลคูณของความถี่ของคำกับค่าความถี่ของเอกสารแบบผกผัน (tf * IDF) โดยสูตรที่ใช้คือ

$$w_{ij} = tf_{ij} * (\log(N/n_i) + 1)$$

โดย N = จำนวนของเอกสารทั้งหมดในฐานข้อมูล

4. แบบใช้กับระบบ OKAPI (Robertson, 1995) เป็นสูตรที่คิดค้นโดย Robertson ที่ใช้กับระบบค้นคืน OKAPI ในการวิจัยของ TREC สูตรที่ใช้คือ

$$w_{ij} = \frac{[tf_{ij} * \log(N/n_i) * (K1 + 1)]}{[K1 * ((1-b) + (b * (filesize_j / avgfilesize))) + tf_{ij}]}$$

โดย $filesize_j$ = ขนาดของแฟ้มเอกสาร j

avgfilesize = ค่าเฉลี่ยขนาดของแฟ้มในฐานข้อมูล

K1 และ b เป็นค่าคงที่ โดย Robertson แนะนำค่า K1 = 2 และ b = 0.75

5. แบบใช้กับระบบสมาร์ท (Singhal, 1997) เป็นรูปแบบสูตรที่ใช้เฉพาะค่าความถี่เอกสารและค่านอร์มัลไลเซชันเท่านั้น โดยตัดค่า IDF ออกซึ่ง Singhal แนะนำว่า IDF ไม่เหมาะกับการให้น้ำหนักคำกับเอกสารแบบยาว

$$w_{ij} = \frac{\log(tf_{ij}) + 1}{0.7 + (0.3 * (filesize_j / avgfilesize))}$$

2.9 ขั้นตอนหาคิวรีที่เหมาะสมที่สุด

ปัญหาของระบบค้นคืนข้อความคือผู้ใช้จะป้อนคิวรีที่ไม่ตรงกับคำที่เอกสารที่ตรงตามต้องการส่วนใหญ่ใช้กัน และโดยมากบางส่วนของเอกสารที่ตรงตามต้องการไม่ถูกดึงออกมาด้วยก็เพราะเอกสารที่ตรงตามต้องการถูกดัชนี ด้วยคำที่ต่างไปจากคิวรีหรือคำที่ใช้ในเอกสารที่ตรงตามต้องการส่วนมาก ปัญหานี้เป็นปัญหาหลักของระบบค้นคืนข้อมูล ซึ่งแสดงถึงความจำเป็นในการปรับปรุงคำถามเพื่อเพิ่มประสิทธิภาพระบบ

การปรับปรุงคิวรีทำเพื่อสร้างคิวรีใหม่ที่สามารถค้นคืนเอกสารที่ตรงตามต้องการ และขจัดเอกสารที่ไม่ตรงตามต้องการ ปัญหาการปรับปรุงคิวรีด้วยผู้ใช้เอง หรือเลือกจากบัญชีคำ คือการควบ

คุมยาก เนื่องจากการปรับปรุงคิวรีให้ถูกเป็นเรื่องไม่ง่ายและยังไม่รู้รูปแบบแน่นอนของเอกสารที่ตรงตามต้องการและเอกสารที่ไม่ตรงตามต้องการ

ปัญหาเหล่านี้สามารถแก้ไขด้วยวิธีการให้ระบบเรียนรู้ความสนใจของผู้ใช้ โดยใช้กลไกการค้นคืนย้อนกลับผ่านผลการค้นคืนเริ่มต้น คือให้ผู้ใช้เลือกว่าเอกสารใดตรงตามต้องการ และเอกสารใดไม่ตรงตามต้องการโดยผู้ใช้จะมีปฏิสัมพันธ์ (interact) กับระบบ ซึ่งเป็นการฝึกฝน (train) ระบบด้วยวิธีการค้นคืนย้อนกลับ

2.10 การค้นคืนย้อนกลับ (Relevance feedback)

หลายครั้งที่คิวรีเริ่มต้นที่ผู้ใช้ป้อนเข้าสู่ระบบค้นคืน ไม่ได้ให้ผลลัพธ์ที่พอใจ ซึ่งอาจเกิดมาจากหลายๆ สาเหตุ เช่น การที่ผู้ใช้มีความรู้เกี่ยวกับเรื่องที่ต้องการน้อยมาก หรือการที่ไม่รู้เนื้อหาที่มีในฐานข้อมูล ซึ่งสิ่งเหล่านี้ก่อให้เกิดการป้อนคิวรีที่มีความหมายกว้างหรือแคบ จนเกินไปกว่าที่ต้องการ ดังนั้นผลลัพธ์จากการค้นคืนจึงมีปริมาณมากหรือน้อยเกินไป จึงทำให้ต้องมีการป้อนคิวรีซ้ำๆอีก จากแนวคิดนี้ ระบบค้นคืนควรจะสามารถช่วยเหลือผู้ใช้ เพื่อให้ประสิทธิภาพของระบบที่ดีขึ้นจึงเกิดวิธีที่เรียกว่า "การค้นคืนย้อนกลับ" ซึ่งเกิดจากการที่ผู้ใช้ต้องป้อนข้อมูลกลับสู่ระบบเพื่อตัดสินใจว่ามีเอกสารใดบ้าง ที่ค้นคืนมาแล้วตรงกับความต้องการบ้าง แล้วระบบจะนำผลลัพธ์ที่ผู้ใช้ป้อนกลับมา ไปช่วยสร้างคิวรีใหม่ที่ดีขึ้น เพื่อให้ผู้ใช้ป้อนคิวรีใหม่กลับสู่ระบบอีก

จุดสำคัญต่อไปในการออกแบบการค้นคืนย้อนกลับคือรูปแบบการเปลี่ยนแปลงคิวรี ซึ่งมีอยู่ด้วยกัน 2 แบบดังนี้คือ

1. การเปลี่ยนแปลงค่าน้ำหนักคำในคิวรี
2. การเปลี่ยนแปลงคำที่ใช้ในคิวรี

การเปลี่ยนแปลงค่าน้ำหนักคำในคิวรี เป็นวิธีที่ช่วยให้ค่าน้ำหนักของคำที่อยู่ในเอกสารที่ตรงตามต้องการเพิ่มขึ้น และลดค่าน้ำหนักของคำที่อยู่ในเอกสารที่ไม่ตรงตามต้องการ ผลการเปลี่ยนแปลงค่าน้ำหนักคำในคิวรี จะช่วยให้ผลการจัดลำดับเอกสารที่มีคำตรงกับคำที่ใช้ในคิวรีเริ่มต้นเท่านั้น มีลำดับที่ดีขึ้น เช่นเอกสารที่ตรงตามต้องการมีผลการจัดลำดับเริ่มต้นที่ตำแหน่งที่ 1, 4 และ 8 เมื่อเปลี่ยนแปลงค่าน้ำหนักคำในคิวรีแล้วผลการจัดลำดับครั้งใหม่ก็จะเปลี่ยนลำดับไปที่ 1, 2 และ 4 เป็นต้น จะเห็นได้ว่าการเปลี่ยนแปลงค่าน้ำหนักคำในคิวรี ทำได้เพียงช่วยให้ผลการจัดลำดับเอกสารดีขึ้นเท่านั้น เนื่องจากคิวรียังคงใช้คำเดิมในการค้นคืน โดยไม่มีคำใหม่เพิ่มเข้าไป

อีกวิธีหนึ่งคือการเปลี่ยนแปลงคำที่ใช้ในคิวรี ทำให้ได้คำใหม่ซึ่งทำให้ช่วยเพิ่มเอกสารที่ตรงตามต้องการแต่มีคำไม่ตรงกับคำที่ใช้ในคิวรีเริ่มต้นมีโอกาสถูกค้นคืนมากขึ้น ซึ่งสามารถทำได้ 2 วิธีคือ

1. การใช้บัญชีคำ เพื่อหาคำที่สัมพันธ์กับคำที่ใช้ในคิวรีเริ่มต้น โดยจะเป็นคำที่มีความหมายกว้างขึ้น หรือแคบลงกว่าคิวรีเริ่มต้น

2. การค้นคืนย้อนกลับด้วยการเปลี่ยนแปลงคำในคิวรีโดยใช้วิธีวิเคราะห์จากเอกสารที่ถูกเลือกจากผู้ใช้ว่าเป็นเอกสารที่ตรงตามต้องการหรือไม่ตรงตามต้องการ แล้วจึงเลือกคำในเอกสาร ซึ่งสามารถเลือกทุกๆ คำ หรือเลือกเฉพาะคำที่มีค่าน้ำหนักคำสูงสุด การเพิ่มหรือลบคำเหล่านี้ควรต้องมีการตัดสินใจด้วยความระมัดระวัง เนื่องจากสามารถทำให้เกิดผลลัพธ์ที่แย่ลงได้

จะเห็นได้ว่าคำใหม่ที่ได้จากการใช้บัญชีคำเกิดจากการใช้เฉพาะแต่ข้อมูลของคิวรีโดยใช้ข้อมูลเกี่ยวกับความหมายของคำ แต่คำใหม่ที่ได้การค้นคืนย้อนกลับจะใช้ข้อมูลจากเอกสารทั้งหมดโดยใช้ข้อมูลเกี่ยวกับจำนวนคำที่พบ

ฮาร์แมน (Harman, 1992) ได้คิดค้นวิธีขยายคำในคิวรีจากเอกสารที่ตรงตามต้องการ และไม่มีการเปลี่ยนค่าน้ำหนักคำในคิวรี โดยสร้างรายการของคำจากเอกสารที่ตรงตามต้องการนำมาจัดลำดับคำ ซึ่งมีรูปแบบการเลือกคำเพื่อนำมาใช้ขยายคิวรี 2 วิธีคือ

1. ใช้การวิเคราะห์สถิติการเกิดขึ้นของคำในเอกสารที่ตรงตามต้องการ มาเทียบกับที่เกิดขึ้นในเอกสารทั้งหมด แล้วจึงนำรายการคำมาเรียงตามค่าสถิตินั้น

2. ใช้การวิเคราะห์ความถี่ของคำที่ปรากฏในชุดของเอกสารที่ถูกค้นคืนมาได้ และนำมาเรียงตามค่าความถี่ของคำที่เกิดขึ้น

การจัดลำดับรายการคำจะใช้วิธีเหมือนการจัดลำดับที่กล่าวไว้ข้างต้นคือการให้น้ำหนักคำนั่นเอง รายการของคำเหล่านี้แสดงให้ผู้ใช้เลือกคำที่ต้องการ แล้วจึงนำคำเหล่านั้นมาเพิ่มเข้าไปในคิวรี เพื่อทำการค้นคืนซ้ำอีกครั้ง

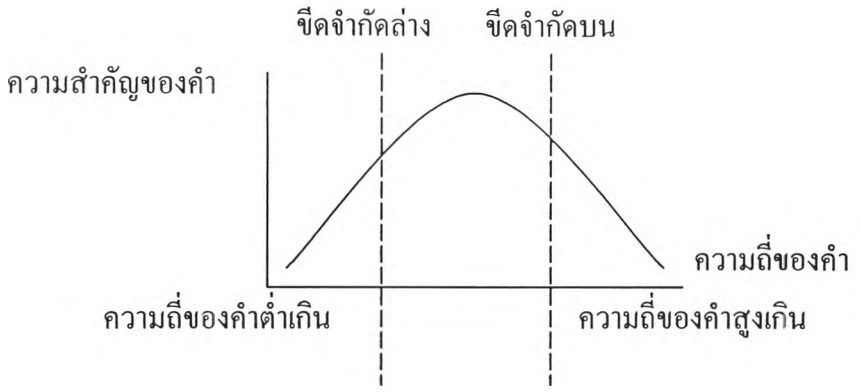
สำหรับรูปแบบการค้นคืนย้อนกลับในงานวิจัยนี้ จะใช้วิธีการค้นคืนย้อนกลับแบบการปรับปรุงคิวรีซึ่งสามารถช่วยเพิ่มเอกสารที่ตรงตามต้องการได้ โดยอาศัยเฉพาะค่าเอกสารที่ตรงตามต้องการจากผู้ใช้เท่านั้น สำหรับการเลือกคำที่จะนำมาใช้ขยายคิวรีต้องมีการคำนวณน้ำหนักคำที่ได้จากเอกสารที่ผู้ใช้ป้อนกลับมาเพื่อแสดงว่าตรงตามต้องการนั้น ทำได้โดย (Robertson, 1990) โดยการคำนวณค่าน้ำหนักคำ (w(i)) ทำได้ดังนี้

$$w(i) = r * \log [((r + 0.5) * (N - n - R + r + 0.5)) / ((n - r + 0.5) * (R - r + 0.5))]$$

โดยที่ r = จำนวนของเอกสารที่รู้ว่าตรงตามต้องการที่มีคำศัพท์ i อยู่
R = จำนวนของเอกสารที่รู้ว่าตรงตามต้องการ
N = จำนวนของเอกสารทั้งหมดในฐานข้อมูล
n = จำนวนของเอกสารทั้งหมดที่มีคำศัพท์ i ในฐานข้อมูล

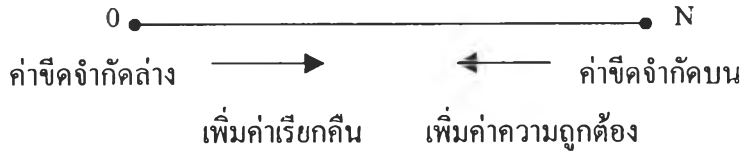
จากสูตรให้น้ำหนักคำข้างต้น จะนำค่าจำนวนเอกสารที่รู้ว่าตรงตามต้องการที่ได้จากผู้ใช้มาใช้ในการคำนวณค่าน้ำหนักของคำด้วย ถ้าให้ r = R = 0 ผลของสูตรคือค่าความถี่เอกสารแบบผกผัน สำหรับการคำนวณน้ำหนักคำทุกคำในเอกสารที่แสดงว่าตรงตามต้องการ ทำให้ระบบใช้เวลาในการประมวลผลนานมาก กว่าจะได้ชุดคำใหม่ให้ผู้ใช้เลือกเพื่อนำไปป้อนกลับสู่คิวรีเริ่มต้น เนื่องจากการคำนวณน้ำหนักคำที่มีความถี่ของคำสูงมากๆ เช่นคำว่า "the" ดังนั้นจึงมีการใช้ค่าขีดจำกัด (Threshold) โดยค่าขีดจำกัดบน (Upper threshold) ใช้เพื่อลดคำที่มีความถี่ของคำสูงเกินค่าขีดจำกัดบนออก และค่าขีดจำกัดล่าง (Lower threshold) ใช้เพื่อลดคำที่มีความถี่ของคำต่ำเกินค่าขีดจำกัดล่างออก โดยสามารถปรับค่าขีดจำกัดทั้งสองได้ ซึ่งทำให้สามารถลดการประมวลผลคำที่ไม่มีประโยชน์ออกไป

ความสัมพันธ์ระหว่างความสำคัญของคำ กับค่าความถี่ของคำแสดงดังรูปที่ 2.5 ให้แกน Y แทนค่าความสำคัญของคำ และแกน X แทนค่าความถี่ของคำ จะเห็นว่ากลุ่มคำที่มีความถี่ของคำสูงคือค่าทางขวาของขีดจำกัดบน และกลุ่มคำที่มีความถี่ต่ำคือค่าทางซ้ายของขีดจำกัดล่าง เป็นกลุ่มคำที่มีค่าความสำคัญต่ำหรือไม่มีความสำคัญสามารถตัดทิ้งได้ ส่วนกลุ่มคำที่มีความถี่ของคำในช่วงกลางคือค่าที่อยู่ระหว่างทางขวาของขีดจำกัดล่างและทางซ้ายของขีดจำกัดบน จะมีค่าความสำคัญสูงสุด สำหรับในระบบค้นคืนสารสนเทศจะสนใจเฉพาะคำที่มีความสำคัญ ดังนั้นสามารถใช้ค่าขีดจำกัดเพื่อตัดกลุ่มคำที่ไม่สำคัญออก เพื่อนำเฉพาะกลุ่มคำที่มีความสำคัญมาใช้ในระบบ



รูปที่ 2.5 กราฟแสดงความสัมพันธ์ระหว่างความสำคัญและความถี่ของคำ

ค่าขีดจำกัดสามารถปรับได้ตามต้องการ โดยผลของการปรับค่าขีดจำกัดดังรูปที่ 2.6 ให้ค่าความถี่ของคำสูงสุดเป็น N ผลการปรับค่าขีดจำกัดล่างเพิ่มขึ้นซึ่งหมายถึงการเลือกคำที่มีค่าความถี่ของคำสูงขึ้น จะทำให้ได้คำใหม่ที่ช่วยเพิ่มค่าเรียกคืนระบบให้สูงขึ้น ส่วนผลของการปรับค่าขีดจำกัดบนลดลงซึ่งหมายถึงการขจัดคำที่มีค่าความถี่สูงออก จะทำให้ได้คำใหม่ที่ช่วยเพิ่มค่าความถูกต้องกับระบบให้สูงขึ้นเช่นกัน ดังนั้นการปรับค่าขีดจำกัดขึ้นอยู่กับรูปแบบความต้องการผลลัพธ์ของการค้นคืนของผู้ใช้แต่ละคน



รูปที่ 2.6 แสดงผลการปรับค่าขีดจำกัด