บทที่ 2
วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

เอกสารเป็นทรัพยากรที่สำคัญอย่างหนึ่งขององค์กรดังนั้นเอกสารที่เกิดจากการดำเนินงานของหน่วยงานจึงเป็นหลักฐานการดำเนินภารกิจและกิจกรรมของหน่วยงานสามารถใช้เป็นเอกสารอ้างอิงเมื่อมีปัญหาทางกฎหมาย การนำเทคโนโลยีสารสนเทศมาประยุกต์ใช้ในการบริหารจัดการเอกสารอิเล็กทรอนิกส์จึงเป็นที่นิยมใช้กันอย่างแพร่หลายเพราะมีประโยชน์หลายด้าน ได้แก่ ลดปัญหาการใช้กระดาษ ประหยัดงบประมาณ ลดขั้นตอนการติดต่อสื่อสารภายในองค์กร และผู้ใช้งานสามารถค้นหาเรียกดูเอกสารได้อย่างรวดเร็ว เพิ่มประสิทธิภาพการทำงาน สามารถบริหารจัดการผ่านระบบเครือข่ายอินเทอร์เน็ตและอินทราเน็ตได้ง่าย

2.1 ความรู้เบื้องต้นเกี่ยวกับกองทุนรวม

กองทุนรวม คือ เครื่องมือในการลงทุน (investment vehicle) สำหรับผู้ลงทุนรายย่อยที่ประสงค์จะนำเงินมาลงทุนในตลาดเงินตลาดทุนแต่ติดขัดด้วยอุปสรรคหลายประการที่ทำให้การลงทุนด้วยตนเองไม่สามารถได้ผลลัพธ์ตามเป้าหมายที่ต้องการ (ธนัยวงศ์ กีรติวานิชย์ และภัสรา ชวาลกร, 2547) เช่น

2.1.1 มีทุนทรัพย์จำนวนจำกัดไม่สามารถกระจายการลงทุนในหลักทรัพย์ต่างประเภทได้มากพอเพื่อลดความเสี่ยงจากการลงทุนหรือ (ธนัยวงศ์ กีรติวานิชย์ และภัสรา ชวาลกร, 2547)

2.2.2 ไม่มีประสบการณ์ความรู้ความชำนาญในการลงทุนหรือ (ธนัยวงศ์ กีรติวานิชย์ และภัสรา ชวาลกร, 2547)

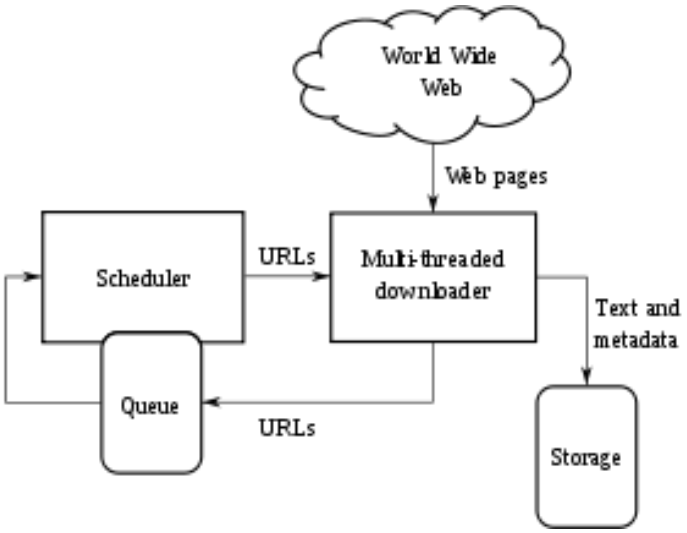
2.2.3 ไม่มีเวลาจะศึกษาค้นหาและติดตามข้อมูลเพื่อใช้ในการตัดสินใจการลงทุน (ธนัยวงศ์ กีรติวานิชย์ และภัสรา ชวาลกร, 2547)

2.2.4 กองทุนรวมจึงเป็นเครื่องมือในการลงทุนที่มีประสิทธิภาพมีการจัดการลงทุนอย่างเป็นระบบโดยมีจุดมุ่งหมายให้การลงทุนได้รับผลตอบแทนที่ดีสุดภายใต้กรอบความเสี่ยงที่ผู้ลงทุนยอมรับได้ (ธนัยวงศ์ กีรติวานิชย์ และภัสรา ชวาลกร:2547) รายละเอียดที่เหลือด้านความรู้เกี่ยวกับกองทุนรวมสามารถอ่านได้ที่ภาคผนวก ก.

2.2 เครื่องมือสืบค้นข้อมูลทางเว็บ

เครื่องมือสืบค้นข้อมูลทางเว็บได้รับการออกแบบมาเพื่อค้นหาข้อมูลบนเวิลด์ไวด์เว็บ และเซิร์ฟเวอร์ FTP ผลการค้นหามักจะได้รับการนำเสนอในรูปแบบรายการผลการค้นหา ที่เรียกกันว่า SERPS หรือ “search engine results pages-หน้าผลการค้นหาจากเครื่องมือสืบค้นข้อมูล” จะประกอบไปด้วยหน้าเว็บ รูปภาพ ข้อมูล และไฟล์ในลักษณะอื่นๆ เครื่องมือสืบค้นข้อมูลบางตัวยังดึงเอาข้อมูลที่มีในฐานข้อมูลหรือ directory เปิด (Open Directory) ไม่เหมือนกับ web directories ที่มีการดูแลรักษาโดยผู้ดูแลที่เป็นมนุษย์เท่านั้น เครื่องมือสืบค้นข้อมูลยังได้รับการดูแลรักษาข้อมูลแบบ real-time โดยการดำเนินการ Algorithm บน web crawler (Web search engine, Retrieved 2012, January23)

2.2.1 เครื่องมือค้นหาเว็บไซต์ทำงานอย่างไร



ภาพที่ 2.1 สถาปัตยกรรมขั้นสูงของ Web Crawler มาตรฐาน

ที่มา: Wikipedia (http://en.wikipedia.org/wiki/Web_search_engine)

เครื่องมือค้นหานั้นทำงานเป็นลำดับดังนี้:

1. Web Crawling
2. Indexing
3. Searching

เครื่องมือค้นหาเว็บไซต์ทำงานโดยการเก็บข้อมูลเกี่ยวกับหน้าเว็บต่างๆ หลายหน้า ซึ่งได้มาจาก html นั่นเอง เพจเหล่านี้ถูกดึงมาโดย Web Crawler (บางครั้งก็รู้จักกันในนาม “แมงมุม”) ซึ่งเป็น Web Browser อัตโนมัติที่ติดตามการเชื่อมโยง (ลิงค์) ทุกอย่างในเว็บไซต์ การจะยกเว้นการเชื่อมโยงใด ๆ ทำได้โดยการใช้ robots.txt เนื้อหาในแต่ละหน้าสามารถนำมาวิเคราะห์เพื่อระบุว่าจะจัดทำเป็นดรรชนี (index) อย่างไร (ตัวอย่างเช่น คำต่างๆ ถูกดึงออกมาจากชื่อเรื่อง หัวเรื่อง หรือส่วนพิเศษต่างๆ เรียกว่า meta tags) ข้อมูลเกี่ยวกับเว็บเพจนั้นถูกเก็บในฐานข้อมูลดรรชนีเพื่อใช้ในอนาคต คำถามที่ตั้งสำหรับการค้นหานั้นอาจจะเป็นคำเดียวโดดๆ ก็ได้ วัตถุประสงค์ของดรรชนี (index) ก็เพื่อให้ข้อมูลนั้นได้รับการค้นเจอเร็วที่สุดเท่าที่จะทำได้ เครื่องค้นหาข้อมูลบางตัว เช่น Google นั้นเก็บหน้าที่เป็นแหล่งข้อมูล (source) (เรียกว่า cache) ไว้ทั้งหมดหรือบางส่วน เช่นเดียวกับข้อมูลที่เกี่ยวข้องกับ web page ในขณะที่ตัวอื่นๆ เช่น AltaVista นั้นเก็บทุกคำของทุกหน้าที่มันค้นพบ หน้าเพจ cache นี้จะเก็บตัวอักษรที่ค้นหาตัวจริงไว้ตลอดเวลา เนื่องจากว่า มันเป็นตัวที่ถูกทำดรรชนี ดังนั้น มันจึงเป็นประโยชน์มากเมื่อเนื้อหาของหน้าปัจจุบันได้รับการปรับเปลี่ยนให้เป็นปัจจุบัน และคำที่ใช้ค้นหานั้นหายไปแล้ว ปัญหานี้อาจจะได้รับการพิจารณาว่าเป็นปัญหาเล็กๆ สำหรับ linkrotและการจัดการของ Google ได้เพิ่มความสามารถในการใช้งานโดยการทำให้ผู้ใช้พึงพอใจเพราะว่าคำที่ใช้ค้นหานั้นปรากฏอยู่ที่หน้าเว็บที่ปรากฏผลออกมา สิ่งนี้บรรลุหลักการประหลาดใจน้อยที่สุด เนื่องจากผู้ใช้มักจะคาดหวังให้คำที่ใช้ค้นหานั้นปรากฏอยู่ในหน้าที่ปรากฏผลออกมาด้วย ความเกี่ยวข้องในการค้นหาที่เพิ่มมากขึ้นทำให้หน้าเว็บ cache เหล่านี้มีประโยชน์มาก นอกเหนือจากประโยชน์ที่หน้าเหล่านี้ มีข้อมูลที่ไม่สามารถหาได้จากที่อื่น (Web search engine, Retrieved 2012, January23)

เมื่อผู้ใช้ใส่คำค้นหาลงไปในเครื่องมือค้นหา (โดยปกติแล้วจะใส่ คำสำคัญลงไป) เครื่องมือค้นหาจะตรวจสอบดรรชนีของมันเอง และแสดงรายการหน้าเว็บที่มีความเกี่ยวข้องกับคำที่ค้นหามากที่สุดตามเกณฑ์ โดยปกติแล้วจะมีสรุปย่อสั้นๆ ที่บ่งบอกชื่อของเอกสาร และบางครั้งก็จะแสดงข้อความบางส่วนด้วย ดรรชนีนี้สร้างขึ้นจากข้อมูลที่เก็บไว้ และผ่านวิธีการนำข้อมูลไปทำดรรชนี เป็นที่น่าเสียดายว่า ยังไม่มีเครื่องมือค้นหาสาธารณะใดที่อนุญาตให้ค้นหาเอกสารได้โดยวันที่ เครื่องมือค้นหาต่างๆ สนับสนุนการใช้งาน ตัวดำเนินการ Boolean และ หรือ และไม่ระบุคำค้นหาเพิ่มเติม ตัวดำเนินการ Boolean นั้นก็เพื่อการค้นหาคำตามตัวอักษร ที่อนุญาตให้ผู้ใช้สามารถเพิ่มการค้นหาให้ละเอียดยิ่งขึ้นและขยายผลการค้นหา เครื่องมือค้นหาจะค้นหาคำหรือวลีตรงตามตัวอักษรที่พิมพ์เข้าไป เครื่องมือค้นหาบางเครื่องมีเครื่องมือระดับสูงที่เรียกว่า การค้นหา ใกล้ชิด ที่อนุญาตให้ผู้ใช้ระบุระยะห่างระหว่างคำสำคัญต่างๆ ยังมีการค้นหาที่อยู่บนพื้นฐานของแนวคิด ที่การค้นหามักจะต้องมีการใช้การวิเคราะห์เชิงสถิติ บนหน้าต่างๆ ที่มีคำหรือวลีที่คุณค้นหาอยู่

นอกจากนี้ยังมีการช่วยให้ผู้ใช้สามารถใช้คำพูดเหมือนที่ใช้ถามมนุษย์ด้วยกันในการพิมพ์ลงในช่องเพื่อการค้นหา เว็บไซต์ค้นหาเช่นนี้ คือ ask.com (Web search engine, Retrieved 2012, January23)

ประโยชน์ที่ได้จากเครื่องมือค้นหานั้นขึ้นอยู่กับความเกี่ยวข้องของชุดผลการค้นหาที่ได้รับ แม้ว่าจะมีหน้าเว็บเป็นล้านๆ ที่มีคำหรือวลีเฉพาะ บางหน้าหน้าก็อาจจะเกี่ยวข้องมากกว่าได้รับความนิยมมากกว่า หรือมีอำนาจมากกว่าหน้าอื่นๆ เครื่องมือค้นหาส่วนใหญ่ใช้วิธีการจัดอันดับผลที่ได้เพื่อให้ผลที่ “ดีที่สุด” ขึ้นมาก่อน ส่วนคำถามที่ว่าเครื่องมือค้นหาตัดสินใจอย่างไรว่าหน้าไหนตรงกับคำค้นหามากที่สุด และลำดับใดที่ผลการค้นหาควรจะได้รับการแสดงออกมานั้นเปลี่ยนแปลงไปตามเครื่องมือค้นหาแต่ละอัน วิธีการยังเปลี่ยนแปลงไปตามเวลาเนื่องจากการใช้อินเตอร์เน็ตนั้นเปลี่ยนแปลงไปและเทคนิคใหม่ๆ ได้รับการพัฒนาขึ้น มีเครื่องมือค้นหาสองแบบหลักๆ ที่เกิดขึ้น อย่างแรกคือระบบที่มีการให้คำจำกัดความและจัดอันดับคำสำคัญต่างๆไว้เรียบร้อยแล้วโดยมนุษย์ อีกแบบคือระบบที่สร้าง “ดรรชนีสลับ-inverted index” โดยการวิเคราะห์ข้อความตามตำแหน่ง แบบที่สองนี้อาศัยคอมพิวเตอร์เป็นอย่างมากในการทำงานส่วนใหญ่ (Web search engine, Retrieved 2012, January23)

เครื่องมือค้นหาเว็บไซต์ส่วนใหญ่เป็นธุรกิจ ที่ได้รับการสนับสนุนโดยรายได้จากการโฆษณา ดังนั้น บางเครื่องมือจึงอนุญาตให้ผู้โฆษณา จ่ายเงินเพื่อให้รายการของพวกเขาอยู่ในอันดับที่สูงกว่าในผลการค้นหา เครื่องมือค้นหาที่ไม่รับเงินสำหรับผลการค้นหานั้น หารายได้โดยการดำเนินการค้นหาการโฆษณาที่เกี่ยวข้อง ไปพร้อมๆกับผลการค้นหาตามปกติ เครื่องมือค้นหาจะได้รายได้ทุกครั้งที่มีคนกดลิงค์โฆษณา (Web search engine, Retrieved 2012, January23)

2.3 Web crawler

Web Crawler เป็นโปรแกรมคอมพิวเตอร์ ที่ค้นดู World Wide Web อย่างเป็นขั้นตอนและอัตโนมัติ หรือเป็นลำดับ คำที่ใช้เรียก Web Crawler อื่นๆ ได้แก่ ants, automatic indexers-ตัวสร้างดรรชนีอัตโนมัติ, bots, Web Spiders, Web Robots หรือ ที่ใช้กันในชุมชน FOAF คือ Web Scutters (Web crawling, Retrieved 2012, January23)

กระบวนการนี้ เรียกว่า Web Crawling หรือ Spidering เว็บไซต์ต่างๆ มากมาย โดยเฉพาะอย่างยิ่ง เครื่องมือค้นหา ใช้การ spidering เป็นวิธีการปรับปรุงข้อมูลให้ทันสมัยอยู่เสมอ Web Crawlers มักจะถูกใช้ในการทำสำเนาหน้าเว็บที่เคยเข้าไปทั้งหมด เพื่อการดำเนินการโดยเครื่องมือค้นหาในอนาคตที่จะจัดระเบียบ จัดทำดรรชนีหน้าเว็บที่ดาวน์โหลดลงมาแล้ว เพื่อให้สามารถค้นหาได้อย่างรวดเร็ว Crawlers ยังสามารถนำไปใช้เพื่อการดูแลรักษางานต่างๆ บนเว็บไซต์โดยอัตโนมัติ อาทิ การตรวจสอบลิงค์ (การเชื่อมโยง) หรือการยืนยันรหัส HTML

นอกจากนั้น Crawlers ยังสามารถนำไปใช้ในการรวบรวมข้อมูลเฉพาะประเภทต่างๆ จากหน้าเว็บ เช่นการค้นหาที่อยู่ e-mail (โดยปกติมักจะใช้ในการส่งจดหมายขยะ-spam) (Web crawling, Retrieved 2012, January23)

Web Crawler เป็น bot หรือเป็นตัวแทนซอฟต์แวร์ (software agent) ประเภทหนึ่ง โดยทั่วไปจะเริ่มด้วยรายการ URLs ที่จะไปค้นดู เรียกว่า seeds เมื่อ crawler เข้าเยี่ยมชม URLs ต่างๆ มันจะระบุ การเชื่อมโยงหลายมิติ (Hyperlinks) ทั้งหมดในหน้านั้นและเพิ่มการเชื่อมโยงเหล่านั้นลงไปใน รายการ URLs ที่จะไปค้นดู เรียกว่า Crawl frontier ซึ่ง URLs ต่างๆจาก frontier นั้นจะถูกเข้าไปค้นดูแบบ recursive โดยจะเป็นไปตามชุดนโยบายที่กำหนด (Web crawling, Retrieved 2012, January23)

ปริมาณที่มากนั้นบ่งชี้ว่า crawler สามารถดาวน์โหลดเศษเสี้ยวหนึ่งของเว็บไซต์ในเวลาที่กำหนดเท่านั้น ดังนั้นมันจะต้องเรียงลำดับความสำคัญของรายการที่จะดาวน์โหลด อัตราการเปลี่ยนแปลงที่สูงนั้นสะท้อนว่าหน้าเว็บต่างๆ อาจจะได้รับการปรับปรุงให้ทันสมัยแล้วหรืออาจจะถูกลบไปแล้วก็ได้ (Web crawling, Retrieved 2012, January23)

จำนวน URLs ที่สามารถเข้าไปค้นดูโดย Crawler ที่ถูกสร้างขึ้นโดยซอฟต์แวร์ด้านเซิร์ฟเวอร์นั้นสร้างความยากลำบากให้กับ Web Crawlers ในการหลีกเลี่ยงการจัดเก็บเนื้อหาที่ซ้ำซ้อนกัน มีชุดตัวแปร HTTP GET (บนพื้นฐานของ URL) มากมาย แต่มีเพียงแค่หยิบมือหนึ่งเท่านั้นที่จะให้เนื้อหาที่ไม่เหมือนใครออกมา ตัวอย่างเช่น Gallery รูปภาพออนไลน์ อาจจะเสนอตัวเลือก 3 ตัวเลือกให้กับผู้ใช้ ดังที่ระบุผ่านตัวแปร HTTP GET ใน URL หากมีวิธีการคัดแยกรูปภาพ 4 วิธี มีสามวิธีที่เป็นการดูรูป Thumbnail สองรูปแบบไฟล์ และอีกหนึ่งตัวเลือกเพื่อการหยุดการใช้เนื้อหาที่ผู้ใช้ป้อนเข้ามา เนื้อหาชุดเดียวกันจะสามารถเข้าถึงได้ด้วย 48 URLs ทุก URL อาจจะเชื่อมโยงมาที่เว็บไซต์ การจัดกลุ่มเชิงคณิตศาสตร์นี้ สร้างปัญหากับ Crawler เนื่องจากพวกมันจะต้องจัดเรียงกลุ่มต่างๆ ที่มีมากมาย ที่มีการเปลี่ยนแปลงเพียงเล็กน้อย เพื่อให้ได้มาซึ่งเนื้อหาที่ไม่ซ้ำใคร (Web crawling, Retrieved 2012, January23)

ดังที่ Edwards *et al.*ได้กล่าวไว้ “เมื่อ Bandwidth สำหรับการดำเนินการ Crawl นั้นจำกัดและไม่ใช่ของฟรี มันจึงจำเป็นที่จะต้อง Crawl เว็บต่างๆ อย่างมีประสิทธิภาพ ไม่คำนึงถึงปริมาณที่สามารถเข้าค้นดูได้เป็นหลัก หากยังต้องการรักษาคุณภาพหรือความสดใหม่ให้คงอยู่” Crawler จะต้องคัดเลือกหน้าเว็บที่จะเข้าค้นดูในครั้งต่อไปอย่างระมัดระวัง (Web crawling, Retrieved 2012, January23)

พฤติกรรมของ Web Crawler เป็นผลมาจากการผสมผสานระหว่างนโยบายต่างๆ:

2.3.1 นโยบายการคัดเลือก

นโยบายการคัดเลือก ที่บ่งบอกว่าให้ดาวน์โหลดหน้าเว็บใด เนื่องจากขนาดของเว็บที่มีขอบเขตกว้างในปัจจุบัน แม้แต่เครื่องมือค้นหาขนาดใหญ่ยังครอบคลุมเพียงแค่ส่วนหนึ่งของเว็บไซต์ที่เปิดสู่สาธารณะ งานศึกษา ปี ค.ศ.2005 แสดงให้เห็นว่าเครื่องมือค้นหาใหญ่ๆ จัดทำดรรชนีได้ไม่เกิน 40-70% ของเว็บที่จัดทำดรรชนีได้ งานศึกษาก่อนหน้าโดย Dr.Steve Lawrence และ Lee Giles แสดงให้เห็นว่า ไม่มีเครื่องมือค้นหาใดที่จัดทำดรรชนีได้มากกว่า 16% ของเว็บไซต์ในปี ค.ศ.1999 เนื่องจาก Crawler ดาวน์โหลดเพียงแค่เศษเสี้ยวหนึ่งของหน้าเว็บต่างๆ จึงเป็นสิ่งที่ดีถ้าเศษเสี้ยวที่ดาวน์โหลดลงมานั้นมีหน้าเว็บที่เกี่ยวข้องมากที่สุด ไม่ใช่เพียงตัวอย่างสุ่มของเว็บไซต์ (Web crawling, Retrieved 2012, January23)

การจะทำเช่นนี้ได้ ต้องการการให้ความสำคัญกับการจัดลำดับความสำคัญของหน้าเว็บต่างๆ ความสำคัญของแต่ละหน้า เป็นฟังก์ชันของคุณภาพของตัวหน้าเว็บ ความนิยมในทำนองของการเชื่อมโยงหรือการเข้าชม และแม้แต่ URL ของมันเอง (อันสุดท้ายนี้เป็นกรณีของเครื่องมือค้นหาแนวดิ่งที่ถูกจำกัดอยู่ในโดเมนขั้นบนสุด (top-down domain) โดเมนเดียว หรือเครื่องมือค้นหาที่ถูกจำกัดอยู่ที่เว็บไซต์เว็บเดียว) การออกแบบนโยบายการคัดเลือกที่ดีนั้นเพิ่มความยากขึ้นมา เพราะว่ามันจะต้องทำงานร่วมกับข้อมูลเพียงส่วนเดียว เนื่องจากชุดหน้าเว็บสมบูรณ์นั้นไม่สามารถรู้ได้ ในระหว่างการ Crawling (Web crawling, Retrieved 2012, January23)

Cho et al. ได้ทำการศึกษานโยบายต่างๆ สำหรับการจัดตารางการ Crawling เป็นครั้งแรก ชุดข้อมูลของพวกเขา คือ หน้าเว็บ 180,000 หน้า ที่ crawl มาจากโดเมนของ standford.edu ซึ่งมีการจำลองการ crawling ด้วยวิธีการที่แตกต่างกันไป Ordering Metrics ที่ได้รับการทดสอบนั้น อยู่ในรูปแบบของ Breadth-first, backlink-count และมีการคำนวณแบบเรียงลำดับหน้า (Pagerank) เป็นบางส่วน หนึ่งในการข้อสรุปคือหาก crawler ต้องการดาวน์โหลดหน้าเว็บที่มีค่า Pagerank สูงก่อนเวลา ในช่วงที่มีการ Crawling นั้น การใช้วิธีการ Pagerank บางส่วนเป็นวิธีที่ดีกว่า ตามมาด้วย breadth-first และ backlink-count อย่างไรก็ตามผลที่ได้นี้เป็นผลสำหรับโดเมนเดียวเท่านั้น Cho ยังเขียนวิทยานิพนธ์ปริญญาเอกของเขาที่ Standford เกี่ยวกับ Web Crawling อีกด้วย (Web crawling, Retrieved 2012, January23)

Najork and Wiener ได้ดำเนินการ crawl จริงๆ กับหน้าเว็บ 328 ล้านหน้าโดยใช้การจัดอันดับแบบ Breadth-first พวกเขาพบว่าการ crawl แบบ breadth-first นั้นจัดเก็บหน้าต่างๆที่มีค่า Pagerankสูงได้ตั้งแต่เนิ่นๆ ในการ crawl (แต่พวกเขาไม่ได้เปรียบเทียบวิธีการนี้กับวิธีการอื่นๆ)

คำอธิบายที่ให้โดยผู้เขียนสำหรับผลที่ได้นี้ก็คือ “หน้าที่มีความสำคัญมากที่สุด มีการเชื่อมโยง (ลิงค์) มากมายมาที่หน้าเหล่านี้โดยโฮสต์ต่างๆมากมาย และจะพบการเชื่อมโยงเหล่านี้ได้แต่เนิ่นๆ โดยไม่คำนึงว่าโฮสต์หรือหน้าเว็บใดที่การ Crawl เริ่มต้นขึ้น (Web crawling, Retrieved 2012, January23)

Abiteboul ได้ออกแบบวิธีการ crawling บนพื้นฐานของ algorithm ที่เรียกว่า OPIC (On-line Page Importance Computation- การคำนวณความสำคัญของหน้าเว็บออนไลน์) ใน OPIC นั้น หน้าเว็บแต่ละหน้าจะได้รับผลรวมชั้นต้นของ “cash” ที่กระจายเท่าๆ กันไปตามหน้าเว็บต่างๆ ที่มีการบ่งชี้ไป คล้ายๆกับการคำนวณ Pagerank แต่เร็วกว่าและดำเนินการในขั้นตอนเดียว Crawler ที่ทำงานโดยใช้ OPIC ดาวน์โหลดหน้าเว็บต่างๆใน Crawling Frontier ที่มีปริมาณ 'cash' สูงก่อนเป็นอันดับแรก การทดลองนั้นดำเนินการกราฟสังเคราะห์ 100,000 หน้า ด้วยการแจกแจง ตามกฎเลขยกกำลัง (power-law) ของ in-links อย่างไรก็ตาม ไม่มีการเปรียบเทียบกับวิธีการอื่นๆ และไม่มีการทดลองบนเว็บจริง (Web crawling, Retrieved 2012, January23)

Boldi et al. ได้ใช้การจำลองกับสับเซ็ตของเว็บ 40 ล้านหน้า จากโดเมน .it และ 100 ล้านหน้าจาก WebBase crawl และทำการทดสอบ breadth-first กับ depth-first ทดสอบการจัดอันดับสุ่มและวิธีการรอบรู้ การเปรียบเทียบนั้นอยู่บนพื้นฐานประสิทธิภาพของการคำนวณค่า PageRank แท้จริงโดยประมาณ ในการ crawl บางส่วน เป็นที่น่าประหลาดใจว่าการค้นดูบางเว็บที่สะสม PageRank อย่างรวดเร็วนั้น (ที่มีชื่อคือการค้นดูแบบ breadth-first และรอบรู้) ให้ค่าการคาดการณ์ก้าวหน้าที่ต่ำมาก (Web crawling, Retrieved 2012, January23)

Baeza-Yates et al. ได้ใช้การจำลองกับสับเซ็ต สองสับเซ็ตของเว็บ 3 ล้านหน้า จากโดเมน .gr และ .cl ทดสอบวิธีการ crawling หลายวิธี พวกเขาได้แสดงให้เห็นว่าทั้งวิธี OPIC และวิธีที่ใช้ความยาวของการเข้าแถวต่อเว็บนั้นดีกว่า การ crawl แบบ breadth-first และยังเป็นวิธีที่มีประสิทธิภาพด้วยในการใช้ crawl ที่ผ่านมาในการแนะแนวให้กับ crawl ปัจจุบัน หากทำได้ (Web crawling, Retrieved 2012, January23)

Daneshpajouh et al. ได้ออกแบบ algorithm บนพื้นฐานของชุมชน สำหรับการค้นหา seeds ที่ดี วิธีการนี้ จะทำการ crawl หน้าเว็บที่มี PageRank สูงจากหลายๆชุมชน ในรูปแบบที่ไม่ซ้ำกัน ในการเปรียบเทียบกับ crawl ที่เริ่มจาก seeds สุ่ม seed ที่ดีจะถูกดึงมาได้จากเว็บที่มีการ crawl ไปก่อนหน้า เมื่อมีการใช้วิธีการใหม่นี้ หากมีการใช้ seeds เหล่านี้ crawl แบบใหม่ก็มีโอกาสที่จะมีประสิทธิภาพสูง (Web crawling, Retrieved 2012, January23)

2.3.1.1 การCrawling แบบเน้นความสนใจ

ความสำคัญของหน้าเว็บสำหรับ Crawler อาจจะแสดงออกมาได้ในรูปแบบฟังก์ชันของความคล้ายคลึงกันของหน้า กับคำที่ใช้ในการค้นหา Web Crawler ที่พยายามจะดาวน์โหลดหน้าต่างๆที่มีความคล้ายคลึงกันนั้นเรียกว่า Crawler เน้นความสนใจ หรือ crawler เฉพาะเรื่อง หลักการของการ Crawling แบบเน้นความสนใจหรือเฉพาะเรื่องนี้ เริ่มมีการพูดถึงโดย Menczer และโดย Chakrabart et al. (Web crawling, Retrieved 2012, January23)

ปัญหาหลักของการ Crawling แบบเน้นความสนใจก็คือ ในบริบทของ Web Crawler นั้น เราต้องการที่จะสามารถพยากรณ์ความคล้ายคลึงกันของเนื้อหาข้อความในหน้าเว็บกับคำค้นหาได้ก่อนที่จะดาวน์โหลดหน้านั้นมาจริงๆ ผู้พยากรณ์ที่เป็นไปได้คือ anchor text ของการเชื่อมโยง (ลิงค์) ต่างๆ นี่เป็นแนวทางที่ใช้โดย Pinkerton ใน web crawler ตัวแรกๆ ของช่วงแรกๆ ของเว็บไซต์ Diligenti et al. เสนอให้มีการใช้เนื้อหาสมบูรณ์ของหน้าต่างๆที่ได้เคยค้นดูมาแล้ว ในการบ่งชี้ความคล้ายคลึงกันระหว่างคำค้นหากับหน้าต่างๆที่ยังไม่เคยค้นดู การทำงานของ Crawling แบบเน้นความสนใจนี้มักจะขึ้นอยู่กับจำนวนของการเชื่อมโยง (ลิงค์) ในหัวข้อเฉพาะที่กำลังถูกค้นหา และ Crawling แบบเน้นความสนใจมักจะอาศัยเครื่องมือค้นหาเว็บทั่วไปในการเริ่มต้น (Web crawling, Retrieved 2012, January23)

1) การจำกัดการเชื่อมโยงที่มีการติดตามแล้ว

Crawler อาจจะต้องการหาหน้า HTML และหลีกเลี่ยงประเภท MIME อื่นๆ ในการที่จะขอทรัพยากร HTML เท่านั้น Crawler อาจจะต้องใช้คำขอ HTTP HEAD เพื่อระบุประเภท MIME ของทรัพยากรเว็บไซต์ก่อนที่จะขอทรัพยากรทั้งหมดด้วยคำขอ GET เพื่อหลีกเลี่ยงการขอ HEAD มากเกินไป Crawler อาจจะตรวจสอบ URL และขอทรัพยากรก็ต่อเมื่อ URL นั้นลงท้ายด้วยอักขระ เช่น .html, .htm, .asp, .aspx, .php, .jsp, .jspxหรือ เครื่องหมาย/ วิธีการนี้อาจจะทำให้ทรัพยากรเว็บ HTML มากมายตกหล่นไปโดยไม่ตั้งใจ (Web crawling, Retrieved 2012, January23)

Crawler บางตัวอาจจะหลีกเลี่ยงการขอทรัพยากรใดๆ ที่มีเครื่องหมาย ?อยู่ด้วย (เกิดขึ้นอย่าง dynamic- มีการทำให้เกิดขึ้น-ผู้แปล) เพื่อที่จะหลบกับดักแมงมุม (Spider Traps) ที่อาจทำให้ crawler ดาวน์โหลด URL จากเว็บไซต์มาอย่างไม่จำกัด วิธีการนี้ไม่น่าเชื่อถือหากเว็บไซต์ใช้การเขียน URL ใหม่เพื่อทำให้ URLต่างๆง่ายขึ้น (Web crawling, Retrieved 2012, January23)

2) การทำURL ให้เป็นมาตรฐาน

Crawler มักจะดำเนินการทำ URL ให้เป็นมาตรฐานอย่างใดอย่างหนึ่งเพื่อหลีกเลี่ยงการ Crawling ทรัพยากรที่เหมือนกันมากกว่าหนึ่งครั้ง ศัพท์คำว่า *การทำ URL ให้เป็นมาตรฐาน* ยังถูกเรียกว่า การเลือก URL ที่ดีที่สุด (URL Canonicalization)ได้อีกด้วย ซึ่งจะหมายถึงกระบวนการใน

การปรับเปลี่ยนและทำ URL ให้เป็นมาตรฐานอย่างสม่ำเสมอ มีประเภทการทำให้เป็นมาตรฐานหลายประเภท ที่อาจจะใช้ดำเนินการ นี่รวมถึงการแปลง URL ให้อยู่ในรูปตัวพิมพ์เล็ก การกำจัดส่วน “.”และ “..” ออก และการเพิ่มเครื่องหมาย / กับส่วนที่ไม่เป็นที่ว่าง เพื่อประโยชน์ในการติดตามค้นหา (Web crawling, Retrieved 2012, January23)

3) การCrawling แบบ Path-ascending (ขึ้นตามทาง-ผู้แปล)

Crawler บางตัวตั้งใจจะดาวน์โหลดทรัพยากรให้ได้มากที่สุดเท่าที่จะทำได้จากเว็บไซต์แต่ละแห่ง ดังนั้นจึงเกิด Path-ascending Crawler ขึ้น เพื่อให้เดินทางขึ้นตามทางทุกทางในแต่ละ URL ที่ Crawler นั้นประสงค์จะ crawl ตัวอย่างเช่น เมื่อได้ seed ที่มี URL เป็น http://llama.org/hamster/monkey/page.htmlCrawler จะพยายาม crawl คำว่า /hamster/monkey/, /hamster/, และ /. Cotheyค้นพบว่า Crawler แบบ path-ascending นี้มีประสิทธิภาพสูงในการค้นหาทรัพยากรที่อยู่โดดๆ หรือทรัพยากรที่โดยปกติแล้วจะหาไม่เจอโดยการใช้การเชื่อมโยงภายใน (inbound link) ในการ Crawlingตามปกติ (Web crawling, Retrieved 2012, January23)

Crawler แบบ path-ascending หลายตัวยังเป็นที่รู้จักในนามซอฟต์แวร์เก็บเกี่ยวเว็บไซต์ (Web harvesting software) เพราะว่าพวกมันถูกใช้ในการ “เก็บเกี่ยว” หรือ เก็บรวบรวมเนื้อหาทั้งหมด – อาจจะเป็นอัลบั้มรูป- จากหน้าเว็บนั้นๆ หรือจากโฮสต์ (Web crawling, Retrieved 2012, January23)

2.3.1.2 นโยบายการกลับไปค้นดู

นโยบายการเข้าค้นดูอีกครั้ง ที่บ่งบอกว่า จะตรวจสอบดูการเปลี่ยนแปลงที่เกิดกับหน้าเว็บเมื่อไร เว็บไซต์นั้นเปลี่ยนแปลงไปได้ตลอดเวลา และการ Crawling เศษเสี้ยวหนึ่งของเว็บไซต์เหล่านั้นอาจใช้เวลาหลายสัปดาห์หรือหลายเดือน เมื่อ Web Crawler เสร็จสิ้นการ Crawl แล้ว ก็อาจจะไม่ทันต่อการเกิดขึ้นใหม่ของเว็บไซต์ การปรับข้อมูลให้ทันสมัยและการลบข้อมูลบางอย่างทิ้งไป (Web crawling, Retrieved 2012, January23)

ในมุมมองของเครื่องมือค้นหา มีต้นทุนที่เข้ามาเกี่ยวข้อง เมื่อไม่สามารถตรวจพบเหตุการณ์ต่างๆ เหล่านั้นได้ ทำให้เกิดการทำสำเนาทรัพยากรที่ไม่เป็นปัจจุบัน ฟังก์ชันต้นทุนที่ใช้กันมากที่สุด คือ ความสดใหม่และอายุ (Web crawling, Retrieved 2012, January23)

ความสดใหม่: คือหน่วยวัดทวิภาค ที่ระบุว่าสำเนาที่ทำขึ้น ณ ตรงนั้น ถูกต้องหรือไม่ ความสดใหม่ของหน้า p ในที่เก็บ ณ เวลา t นั้นสามารถให้คำจำกัดความว่า

$$F_p(t) = \begin{cases} 1 & \text{หาก p เท่ากับสำเนา ณ ตรงนั้น ณ เวลา t} \\ 0 & \text{นอกจากนั้น} \end{cases}$$

ภาพที่ 2.2 สูตรคำนวณความสดใหม่ของข้อมูล

ที่มา: Wikipedia (http://en.wikipedia.org/wiki/Web_crawling)

อายุ: เป็นหน่วยวัดที่ระบุว่าสำเนาที่ทำขึ้น ณ ตรงนั้น ล้าสมัยไปมากเพียงใด อายุของหน้า p ในที่เก็บ ณ เวลา t นั้นสามารถให้คำจำกัดความว่า

$$A_p(t) = \begin{cases} 0 & \text{หาก p ไม่ถูกเปลี่ยนแปลง ณ เวลา t} \\ t - & \text{เวลาที่ใช้ในการเปลี่ยนแปลง p} \end{cases}$$

ภาพที่ 2.3 สูตรคำนวณอายุของข้อมูล

ที่มา: Wikipedia (http://en.wikipedia.org/wiki/Web_crawling)

Coffman et al. ได้ศึกษาคำจำกัดความของเป้าหมายของ Web Crawler ที่มีความหมายเท่ากับความสดใหม่ แต่ใช้อีกคำหนึ่ง: พวกเขาเสนอว่า Crawler จะต้องลดเศษเสี้ยวเวลาที่หน้าต่างๆ จะล้าสมัย ลงให้น้อยที่สุด พวกเขายังสังเกตอีกว่า ปัญหาของ Web Crawling สามารถนำมาแสดงในรูปแบบของ multiple-queue (การเข้าหลายแถว) Single-server polling system (ระบบการลงคะแนนเซิร์ฟเวอร์เดี่ยว) ซึ่ง Web Crawler เป็นเซิร์ฟเวอร์และเว็บไซต์ เป็นแถวหรือคิว การเปลี่ยนแปลงหน้าเว็บ คือ การเข้ามาของลูกค้า และ switch-over times (เวลาที่สลับไปมา) คือช่วงเวลาระหว่างการเข้าถึงหน้าเว็บไปจนถึงเว็บไซต์เว็บหนึ่ง ภายใต้รูปแบบนี้ ระยะเวลารอโดยเฉลี่ยสำหรับลูกค้าในระบบลงคะแนน นั้นจะเท่ากับอายุเฉลี่ยสำหรับ Web Crawler (Web crawling, Retrieved 2012, January23)

เป้าหมายของ Crawler คือการรักษาความสดใหม่เฉลี่ยของหน้าต่างๆ ที่เก็บรวบรวมมาให้สูงที่สุดเท่าที่จะทำได้ หรือทำให้อายุเฉลี่ยของหน้าต่างๆ นั้นต่ำที่สุดเท่าที่จะทำได้ เป้าหมายเหล่านี้ ไม่เหมือนกัน: ในกรณีแรก Crawler เพียงสนใจว่ามีหน้าเว็บที่ล้าสมัยกี่หน้า ในขณะที่ใน

กรณีที่สอง Crawler นั้นสนใจว่า สำเนาหน้าเว็บ ณ จุดนั้น มีอายุเท่าไรแล้วCho และ Garcia-Molina ได้ศึกษานโยบายการกลับไปค้นดูง่ายๆ สองนโยบาย (Web crawling, Retrieved 2012, January23)

นโยบายแบบแผนเดียวกัน (Uniform Policy): รวมเอาการกลับเข้าไปค้นดูหน้าเว็บทุกหน้าที่เก็บรวบรวมมาไว้และมีความถี่เดียวกัน ไม่คำนึงถึงอัตราการเปลี่ยนแปลงของหน้าเว็บเหล่านั้น

นโยบายสัดส่วน (Proportional Policy): รวมเอาการกลับเข้าไปค้นดูหน้าเว็บที่เปลี่ยนแปลงถี่กว่า มากกว่า ความถี่ของการเยี่ยมชมนั้น เป็นสัดส่วนแปรผันตรงกับความถี่การเปลี่ยนแปลง (โดยประมาณ)

(ในทั้งสองกรณี ลำดับการ crawling ซ้ำในหน้าเว็บเดิมอาจกระทำแบบสุ่มหรือแบบคำสั่งตายตัว)

Cho and Garcia-Molina ได้พิสูจน์ผลอันน่าประหลาดใจ ในกรณีความสดใหม่เฉลี่ยนั้น นโยบายแบบแผนเดียวกันให้ผลดีกว่านโยบายสัดส่วน ทั้งใน Web Crawl จำลองและ Web Crawl จริง โดยสัญชาติญาณแล้วเหตุผลมีอยู่ว่า เนื่องจาก Web Crawler มีข้อจำกัดในเรื่องของจำนวนหน้าเว็บที่สามารถ Crawl ได้ในกรอบเวลาหนึ่ง 1.)พวกมันจะจัดสรรการ Crawl ใหม่ๆ ให้กับหน้าเว็บที่เปลี่ยนแปลงอย่างรวดเร็ว มากเกินไป ในขณะที่หน้าเว็บที่มีการเปลี่ยนแปลงในความถี่ที่น้อยกว่าถูกละเลย และ 2.)ความสดใหม่ของหน้าเว็บที่เปลี่ยนแปลงอย่างรวดเร็วนั้นมีระยะเวลาที่สั้นกว่าหน้าเว็บที่มีการเปลี่ยนแปลงในความถี่น้อยกว่า พูดง่ายๆ ก็คือ นโยบายสัดส่วนจัดสรรทรัพยากรให้กับการ Crawl หน้าเว็บที่มีการเปลี่ยนแปลงอย่างรวดเร็ว แต่ประสบกับความสดใหม่เฉลี่ยที่น้อยกว่า (Web crawling, Retrieved 2012, January23)

ในการพัฒนาความสดใหม่ Crawler ควรจะมีบทลงโทษกับองค์ประกอบที่เปลี่ยนแปลงบ่อยเกินไป นโยบายการกลับไปค้นดูที่เหมาะสมที่สุดนั้น ไม่ใช่ทั้งนโยบายแบบแผนเดียวกันหรือนโยบายสัดส่วน วิธีการที่เหมาะสมที่สุดในการรักษาระดับความสดใหม่เฉลี่ยให้สูงนั้น คือการไม่สนใจหน้าเว็บที่เปลี่ยนแปลงเร็วเกินไป และวิธีการที่เหมาะสมที่สุดในการทำอายุเฉลี่ยให้ต่ำก็คือการใช้การเพิ่มขึ้นของความถี่การเข้าถึงฝ่ายเดียว (และ sub-linear) เมื่อเทียบกับอัตราการเปลี่ยนแปลงของหน้าเว็บแต่ละหน้า ในทั้งสองกรณี สิ่งที่เหมาะสมที่สุด จะเข้าใกล้นโยบายแบบแผนเดียวมากกว่านโยบายสัดส่วน ดังที่ Coffman et al. สังเกต “ในการที่จะลดเวลาล้าสมัยโดยประมาณลงให้ต่ำที่สุด การเข้าถึงหน้าเว็บใดๆ ควรจะแจกแจงให้เท่าๆ กัน เท่าที่จะทำได้” โดยทั่วไปแล้วคงไม่สามารถหาสูตรที่ชัดเจนสำหรับนโยบายกลับไปค้นดูได้ แต่สามารถจะหาได้ในเชิงตัวเลข เนื่องจากสิ่งเหล่านี้ ขึ้นอยู่กับการแจกแจงการเปลี่ยนแปลงของหน้าเว็บ Cho และ Garcia-Molina แสดงให้เห็นว่าการแจกแจงแบบ exponential นั้นดีต่อการอธิบายการเปลี่ยนแปลงของหน้า

ในขณะที่ Ipeirotise et al. แสดงให้เห็นว่าจะใช้เครื่องมือทางสถิติในการค้นหาตัวแปรที่ส่งผลกระทบต่อการแจกแจงได้อย่างไร สังเกตว่านโยบายการกลับไปค้นดูที่มีการพิจารณา ณ ที่นี้นั้น สมมติให้หน้าเว็บทั้งหมด มีคุณภาพเหมือนกันหมด (“ทุกๆ หน้าในเว็บ มีคุณค่าเท่ากัน”) ซึ่งในความเป็นจริงแล้วไม่ใช่อย่างนั้น ดังนั้นควรจะมีการนำข้อมูลเพิ่มเติมเกี่ยวกับคุณภาพของหน้าเว็บรวมเข้าไปในการวิเคราะห์ด้วยเพื่อให้เกิดนโยบาย Crawling ที่ดียิ่งขึ้น (Web crawling, Retrieved 2012, January23)

2.3.1.3 นโยบายความสุภาพ

นโยบายความสุภาพ ที่บ่งบอกว่าจะหลีกเลี่ยงภาวะ Overload ของเว็บไซต์ต่างๆ ได้อย่างไร Crawler สามารถดึงข้อมูลได้เร็วกว่าและลึกกว่าผู้ค้นหาที่เป็นมนุษย์ ดังนั้น พวกมันจึงสามารถส่งผลกระทบรุนแรงต่อการทำงานของเว็บไซต์ เป็นที่รู้กันดีว่า หาก Crawler เดี่ยวๆ กำลังดำเนินการตามคำขอหลายๆ คำขอต่อวินาที และ/หรือดาวน์โหลดไฟล์ใหญ่ๆ หลายไฟล์ เซิร์ฟเวอร์ก็อาจจะต้องเผชิญกับความยุ่งยากในการดำเนินการตามคำขอจาก Crawler หลายๆ ตัว

ดังที่ Koster ได้สังเกต การใช้ Web Crawlers นั้นเป็นประโยชน์ในงานต่างๆมากมาย แต่ก็มีข้อเสียต่อชุมชน (Web crawling, Retrieved 2012, January23)โดยทั่วไป ต้นทุนของการใช้ Web Crawlers นั้นมีดังนี้:

1) ทรัพยากรเครือข่าย เนื่องจาก Crawlers นั้นต้องการ bandwidth ที่มากพอสมควร และมีระดับการทำงานอย่างคู่ขนานสูง เป็นเวลานาน

2) เซิร์ฟเวอร์ Overload โดยเฉพาะอย่างยิ่งหากความถี่ของการเข้าถึงเซิร์ฟเวอร์นั้นๆสูงเกินไป

3) Crawlers ที่ถูกเขียนขึ้นมาไม่ดี ซึ่งอาจทำให้เซิร์ฟเวอร์หรือเราเตอร์พัง หรืออาจจะดาวน์โหลดหน้าเว็บที่พวกมันไม่สามารถจะดูแลได้ และ

4) Crawlers ส่วนบุคคลที่ หากถูกใช้โดยผู้ใช้หลายๆคน อาจจะก่อกวนเครือข่ายและเซิร์ฟเวอร์เว็บไซต์ได้

ทางออกส่วนหนึ่งของปัญหาเหล่านี้ก็คือ การใช้โปรโตคอลแยก robots (Robots Exclusion Protocol) หรือเรียกอีกอย่างว่า robots.txt protocol สิ่งนี้เป็นมาตรฐานสำหรับผู้ดูแลในการระบุว่าส่วนใดของเซิร์ฟเวอร์เว็บของพวกเขาที่ไม่ควรจะเข้าถึงได้โดย Crawlers มาตรฐานนี้ ไม่รวมการเสนอให้เข้าดูเป็นการภายในของเซิร์ฟเวอร์เดียวกัน แม้ว่าการเข้าถึงเป็นการภายในนี้จะเป็นวิธีการที่มีประสิทธิภาพมากที่สุดในการหลีกเลี่ยง เซิร์ฟเวอร์ Overload เมื่อไม่นานมานี้ เครื่องมือค้นหาเชิงธุรกิจเช่น Ask Jeeves, MSN และ Yahoo นั้นสามารถใช้ตัวแปร “Crawl-delay” เพิ่มเติมได้

แล้ว ในไฟล์ robots.txt เพื่อระบุจำนวนวินาทีที่คำขอถูกทำให้ช้าไป (Web crawling, Retrieved 2012, January23)

ช่วงเวลาระหว่างการเชื่อมต่อที่เสนอมาอันแรกคือ 60 วินาที อย่างไรก็ตาม หากหน้าเว็บต่างๆถูกดาวน์โหลดจากเว็บไซต์ในอัตรานี้ ด้วยหน้าเว็บมากกว่า 100,000 หน้า แม้ว่าจะมีการเชื่อมต่อที่สมบูรณ์ มีค่าความหน่วงเวลาเป็นศูนย์ และมี Bandwidth ไม่จำกัด ยังจะต้องใช้เวลามากกว่าสองเดือนในการดาวน์โหลดเว็บไซต์เหล่านั้น นอกจากนั้น มีเพียงเศษเสี้ยวของทรัพยากรจาก Web Server นั้นเท่านั้นที่จะถูกใช้ นี่ยังใช้ไม่ได้ (Web Search Engine, Retrieved 2012, January23)

Cho ใช้ช่วงเวลา 10 วินาที ในการเข้าถึง และ WIRE Crawler ใช้เวลา 15 วินาทีเป็นค่าปกติ Mercator Web Crawler ดำเนินการตามนโยบายความสุภาพที่ถูกปรับให้เหมาะ: หากใช้เวลา t วินาทีในการดาวน์โหลดเอกสารหนึ่งจาก server ที่กำหนด crawler จะรอ 10t วินาทีก่อนที่จะดาวน์โหลดหน้าต่อไป Dill et al. ใช้เวลา 1 วินาที (Web crawling, Retrieved 2012, January23)

สำหรับผู้ที่ใช้ Web Crawler เพื่อจุดประสงค์ในการค้นคว้า วิจัย จะต้องมีการวิเคราะห์ข้อดี-ข้อเสียอย่างละเอียดกว่านี้ และจะต้องมีการดำเนินการพิจารณาในเรื่องของจรรยาบรรณด้วยในการตัดสินใจว่าจะ Crawl ที่ใดบ้างและจะ Crawl เร็วเท่าไร (Web crawling, Retrieved 2012, January23)

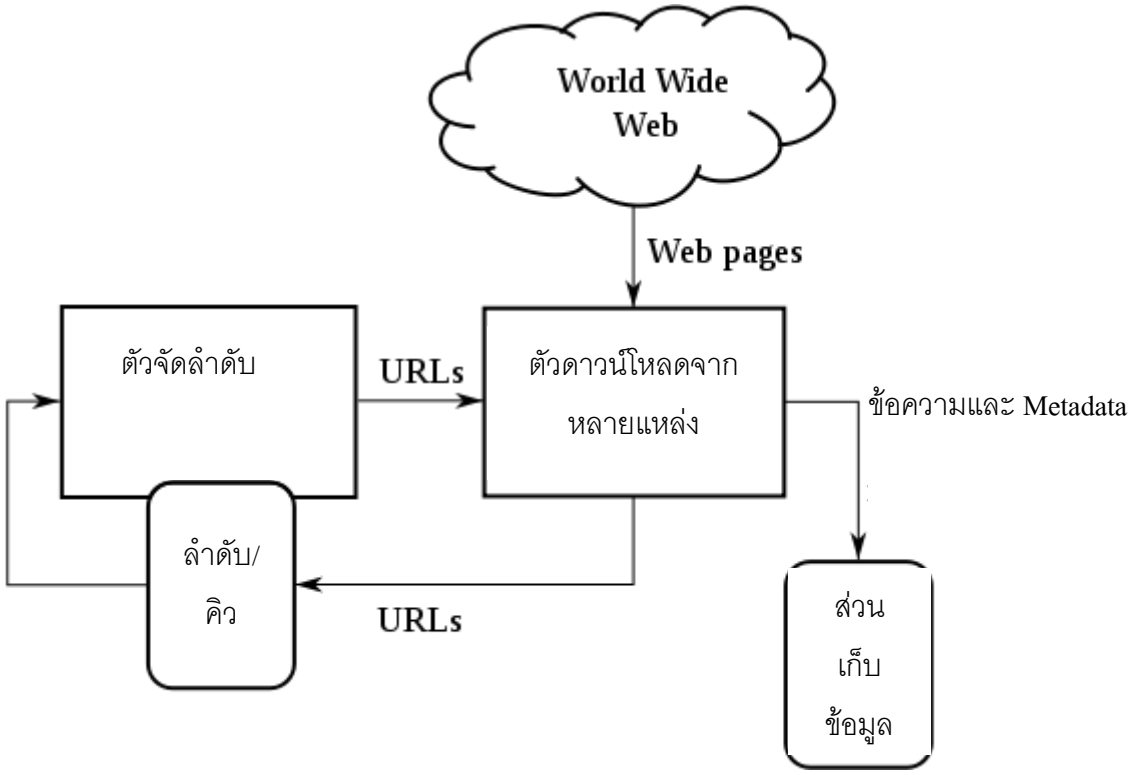
หลักฐานเรื่องราวจากบันทึกการเข้าถึง แสดงให้เห็นว่า ช่วงเวลาในการเข้าถึงจาก Crawlers ที่เป็นที่รู้จักนั้นอยู่ระหว่าง 20 วินาทีและ 3-4 นาที เป็นที่น่าสงเกตว่า แม้ว่าจะสุภาพแล้วและได้ดำเนินการตามมาตรการป้องกันต่างๆแล้วเพื่อหลีกเลี่ยงการ overload เซิร์ฟเวอร์ แต่ก็ยังได้รับคำร้องเรียนจากผู้ดูแลเว็บเซิร์ฟเวอร์ Brin and Page สังเกตว่า “...การใช้งาน Crawler ที่เชื่อมต่อกับเซิร์ฟเวอร์กว่าครึ่งล้าน (...) ทำให้มีอีเมล์และโทรศัพท์เข้ามาว่าจำนวนหนึ่งเลยทีเดียว เนื่องจากมีคนออนไลน์อยู่มาก แน่นอนว่ามีคนที่ไม่รู้ว่า crawler คืออะไร เพราะเป็นครั้งแรกที่พวกเขาเจอกับมัน (Web crawling, Retrieved 2012, January23)

2.3.1.4 นโยบายดำเนินการคู่ขนาน

นโยบายการไม่ทำให้เกิดการขนาน ที่บ่งบอกว่าจะประสานงานกับ Web Crawler ต่างๆที่ส่งออกไปอย่างไร Crawler คู่ขนานคือ Crawler ที่ทำงานหลายๆ กระบวนการขนานกันไป เป้าหมายคือเพื่อให้ได้อัตราดาวน์โหลดสูงที่สุด ในขณะที่ลดค่าใช้จ่ายจากการทำงานคู่ขนานลงให้ต่ำที่สุด และหลีกเลี่ยงการดาวน์โหลดหน้าเดียวกันซ้ำๆ ในการหลีกเลี่ยงการดาวน์โหลดหน้าเดิมมากกว่าหนึ่งครั้งนั้น ระบบการ Crawling ต้องการนโยบายในการระบุ URLs ใหม่ๆ ที่พบเจอ

ระหว่างกระบวนการ Crawling เนื่องจากอาจจะเจอ URL เดียวกันได้อีก ในกระบวนการ Crawling อีกกระบวนการหนึ่ง (Web crawling, Retrieved 2012, January

2.3.2 โครงสร้าง



ภาพที่ 2.5 โครงสร้างขั้นสูงของ Web Crawler พื้นฐาน

ที่มา: Wikipedia (http://en.wikipedia.org/wiki/Web_crawling)

Crawler นั้นไม่เพียงแต่จะต้องมียุทธศาสตร์การ crawling ที่ดีดังที่ได้กล่าวไว้ในก่อนหน้านี้ เท่านั้น แต่ยังต้องมีโครงสร้างที่เหมาะสมเป็นอย่างสูงอีกด้วย

Shkapenyukและ Suelตั้งขอสังเกตว่า (Web crawling, Retrieved 2012, January23)

ในขณะที่การสร้าง crawler ที่ทำงานช้า ดาวน์โหลดไม่กี่หน้าต่อวินาที เป็นเวลาสั้นๆ นั้นค่อนข้างง่าย การสร้างระบบที่มีประสิทธิภาพสูงที่สามารถดาวน์โหลดหน้าเว็บเป็นร้อยล้านหน้าในระยะเวลาหลายสัปดาห์นั้น มีความยากลำบากและเป็นความท้าทายต่อการออกแบบระบบ I/O และประสิทธิภาพของเครือข่าย และความทนทานและความสามารถในการบริหารจัดการได้ (Web crawling, Retrieved 2012, January23)

Web Crawler นั้นเป็นหัวใจสำคัญของเครื่องมือค้นหา และรายละเอียดเกี่ยวกับ algorithm และโครงสร้างนั้นถูกเก็บเป็นความลับทางธุรกิจ เมื่อการออกแบบ Crawler ได้รับการเปิดเผย มักจะมีรายละเอียดที่สำคัญขาดหายไปอยู่บ่อยๆ ซึ่งจะป้องกันไม่ให้คนอื่นลอกเลียนแบบได้ ยังมีความกังวลเกี่ยวกับ “Search Engine Spamming”เกิดขึ้นมา ซึ่งสิ่งนี้จะป้องกันไม่ให้เครื่องมือค้นหาใหญ่ๆ ลงอันดับ algorithm ของพวกเขา (Web crawling, Retrieved 2012, January23)

2.3.3 การระบุตัวตนCrawler

Web Crawler นั้นมักจะระบุตัวตนของตนเองต่อ Web Server โดยการใช้ส่วน User-agent ของคำขอ HTTP ผู้ดูแลเว็บไซต์มักจะตรวจสอบบันทึก web server และใช้ส่วน user agent เพื่อระบุว่า crawler ใดได้เข้าเยี่ยมชมเว็บเซิร์ฟเวอร์ และเข้าชมบ่อยเพียงใด ส่วน user agent นั้นอาจจะใส่ URL ไว้ด้วยซึ่งผู้ดูแลเว็บไซต์อาจจะสามารถเข้าไปหาข้อมูลเพิ่มเติมเกี่ยวกับ Crawler ได้ Spambotsและ Web Crawler ตัวร้ายอื่นๆ มักจะไม่ใส่ข้อมูลระบุตัวตนลงไปใน ส่วน user agent หรือพวกมันอาจจะปิดบังตัวตนที่แท้จริงอยู่ โดยใช้ browser หรือ crawler ที่เป็นที่รู้จักดีตัวอื่นมาบังหน้า (Web crawling, Retrieved 2012, January23)

การที่Web Crawler ระบุตัวตนของตัวเองนั้นเป็นสิ่งสำคัญ เพื่อให้ผู้ดูแลเว็บไซต์สามารถติดต่อเจ้าของได้หากจำเป็น ในบางกรณี Crawler อาจจะถูกกับดักติดอยู่ใน Crawler Trap หรือพวกมันอาจจะกำลัง Overload เซิร์ฟเวอร์ด้วยคำขอต่างๆอยู่และเจ้าของมีความจำเป็นจะต้องหยุด Crawler การแสดงตัวตนนั้นยังมีประโยชน์สำหรับผู้ดูแลที่สนใจที่จะรู้ว่าพวกเขาจะคาดหวังให้หน้าเว็บต่างๆ ได้รับการจัดเข้าดรรชนีโดยเครื่องมือค้นหาต่างๆ เมื่อใด (Web crawling, Retrieved 2012, January23)

2.3.4 ตัวอย่าง Crawler ทั่วไป

รายการต่อไปนี้เป็นรายการโครงสร้างของ Crawler ที่ได้รับการเผยแพร่แล้ว ซึ่งเป็นโครงสร้างสำหรับ Crawler ทั่วไป (ยกเว้น Crawler ที่เน้นเฉพาะเว็บไซต์) พร้อมคำอธิบายสั้นๆ ที่รวมถึงชื่อต่างๆ สำหรับส่วนประกอบต่างๆและลักษณะเด่น:

2.3.4.1 Yahoo! Slurp เป็นชื่อของ Crawler ค้นหาของ Yahoo (Web crawling, Retrieved 2012, January23)

2.3.4.2 Bingbot เป็นชื่อของ Web Crawler ของ Bing แห่ง Microsoft ซึ่งมาแทน MSNbot (Web crawling, Retrieved 2012, January23)

2.3.4.3 FAST Crawler เป็น Crawler ที่กระจายไปทั่ว ถูกใช้งานโดยFast Search & Transfer, และคำอธิบายทั่วไปของโครงสร้าง (Web crawling, Retrieved 2012, January23)

2.3.4.4 Googlebot นั้นได้รับการอธิบายอย่างละเอียด แต่แหล่งอ้างอิงนั้นมีถึงแค่โครงสร้างช่วงแรกๆ ซึ่งอยู่บนพื้นฐานของ C++ และ Python Crawler นั้นได้รับการบูรณาการเข้ากับกระบวนการจัดทำดรรชนี เนื่องจากมีการใช้การวิเคราะห์ข้อความกับการจัดทำดรรชนีข้อความทั้งหมดและใช้การวิเคราะห์ข้อความในการดึงเอา URL ออกมาด้วย มีเซิร์ฟเวอร์ URL ที่ส่งรายการ URL ที่จะต้องถูกดึงมาโดยกระบวนการ Crawling ต่างๆ ระหว่างกระบวนการวิเคราะห์ข้อความ URL ที่พบ ถูกส่งต่อไปยังเซิร์ฟเวอร์ URL ที่ทำการตรวจสอบว่า URL นั้นได้ผ่านตามาก่อนหรือไม่ หากไม่ URL จะถูกเพิ่มเข้าไปในคิวของเซิร์ฟเวอร์ URL (Web crawling, Retrieved 2012, January23)

2.3.4.5 PolyBot เป็น Crawler กระจายไปทั่ว เขียนขึ้นโดยใช้ C++ และ Pythonซึ่งมีส่วนประกอบคือ “ตัวบริหารการ Crawl- Crawl Manager” “ตัวดาวน์โหลด” หนึ่งตัวหรือมากกว่า และ “DNS Resolvers” หนึ่งตัวหรือมากกว่าURL ที่ถูกรวบรวมมาจะถูกเพิ่มเข้าไปในคิวบนดิสก์ และจะถูกดำเนินการต่อไป เพื่อค้นหา URL ที่เคยพบเห็นแล้วในโหมด Batch นโยบายความสุภาพพิจารณาโดเมนทั้งในระดับที่สามและระดับที่สอง (เช่น: www.example.com และ www2.example.com เป็นโดเมนระดับที่สาม) เนื่องจากโดเมนระดับที่สามนั้นมักจะมีโฮสต์เป็นเซิร์ฟเวอร์เว็บตัวเดียวกัน (Web crawling, Retrieved 2012, January23)

2.3.4.6 RBSE เป็น Web Crawler ที่ได้รับการเผยแพร่ตัวแรก มันอยู่บนพื้นฐานของสองโปรแกรม โปรแกรมแรก “spider” นั้นจะรักษาคิว สัมพันธ์กับฐานข้อมูล และโปรแกรมที่สอง “mite” เป็น browser ของ www.ASCIIที่ได้รับการปรับปรุงแล้วที่ดาวน์โหลดหน้าเว็บต่างๆ จากเว็บไซต์ (Web crawling, Retrieved 2012, January23)

2.3.4.7 WebCrawler ใช้งานเพื่อสร้างดรรชนีข้อความทั้งหมดที่เผยแพร่ต่อสาธารณะดรรชนีแรก มันอยู่บนพื้นฐานของ lib-WWW เพื่อใช้ดาวน์โหลดหน้าเว็บและโปรแกรมอื่นเพื่อวิเคราะห์และสั่งการ URLs สำหรับการสำรวจ breadth-first ของ Web Graph มันยังรวมเอา Crawler แบบ real-time ที่จะติดตามลิงค์ (การเชื่อมโยง) ต่างๆ บนพื้นฐานของความคล้ายคลึงกันของข้อความ anchor (anchor text) กับคำถามที่ต้องการค้นหา (Web crawling, Retrieved 2012, January23)

2.3.4.8 World Wide Web Worm เป็น Crawler ที่ถูกใช้ในการสร้างดรรชนีอย่างง่ายของชื่อเอกสาร และ URLs สามารถค้นหาดรรชนีได้โดยการใช้คำสั่ง grepUnix (Web crawling, Retrieved 2012, January23)

2.3.4.9 WebFountain เป็น Crawler กระจายไปทั่ว และเป็น Crawler แบบหน่วย (Modular Crawler) คล้ายๆกับ Mercator แต่ถูกเขียนขึ้นโดยใช้ C++ มีตัว “ควบคุม” ที่ประสานงานกับหน่วย “มด” ต่างๆ หลังจากที่ดาวน์โหลดหน้าต่างๆซ้ำๆกันแล้ว อัตราการเปลี่ยนแปลงจะถูกอนุมานสำหรับหน้าแต่ละหน้า และจะใช้วิธีการ non-linear programming ในการแก้ระบบสมการเพื่อให้ค่าความสดใหม่สูงที่สุด ผู้เขียนแนะนำให้ใช้คำสั่ง Crawling ชนิดนี้ในขั้นตอนแรกๆ ของการ Crawl จากนั้นจึงเปลี่ยนไปใช้ คำสั่ง Crawling ที่เป็นมาตรฐาน ซึ่งหน้าต่างๆ จะได้รับการเข้าค้นดูด้วยอัตราความถี่เท่าๆกัน (Web crawling, Retrieved 2012, January23)

2.3.4.10 WebRACE เป็นหน่วย Crawling และ Caching ที่ใช้งานใน Java และใช้งานเป็นส่วนหนึ่งของระบบ Generic ที่เรียกว่า eRACEระบบนี้จะรับคำขอจากผู้ใช้ในการดาวน์โหลดหน้าเว็บ ดังนั้น Crawler จะทำหน้าที่ส่วนหนึ่งเป็นเซิร์ฟเวอร์Smart Proxy ระบบยังต้องรับคำขอสำหรับการ “subscriptions” กับหน้าเว็บต่างๆที่จะต้องถูกติดตาม: เมื่อหน้าเว็บมีการเปลี่ยนแปลงหน้าเว็บเหล่านั้นจะต้องถูกดาวน์โหลดลงโดย Crawler และตัวรับการ subscription จะต้องได้รับแจ้ง ลักษณะที่โดดเด่นที่สุดของ WebRACEคือ ในขณะที่ Crawler ส่วนใหญ่เริ่มต้นจากชุด URLs “Seed” WebRACEจะได้รับ URLs เริ่มต้นใหม่ๆ อย่างต่อเนื่อง เพื่อใช้ในการเริ่มการ Crawl (Web crawling, Retrieved 2012, January23)

2.3.5 ตัวอย่าง Open-source crawlers

นอกเหนือจากโครงสร้าง Crawler เฉพาะ ในรายการข้างต้นแล้ว ยังมีโครงสร้าง Crawler ทั่วไป ที่เผยแพร่โดย Cho and Chakrabartiอีก (Web crawling, Retrieved 2012, January23)

2.3.5.1 Aspseek เป็น Crawler เป็นตัวจัดทำดรรชนี และเป็นเครื่องมือค้นหาที่เขียนขึ้นใน C++ และจดลิขสิทธิ์ภายใต้ GPL (Web crawling, Retrieved 2012, January23)

2.3.5.2 DataparkSearch เป็น Crawler และเครื่องมือค้นหาที่ออกมาภายใต้ GNU General Public License (ลิขสิทธิ์สาธารณะทั่วไป) (Web crawling, Retrieved 2012, January23)

2.3.5.3 GNU Wget เป็น Crawler ที่ทำงานด้วยสายคำสั่ง เขียนขึ้นใน C และออกสู่สาธารณะภายใต้ GPL มักใช้ในการ mirror เว็บและ FTP sites (Web crawling, Retrieved 2012, January23)

2.3.5.4 GNU Wget เป็น Crawler ที่ทำงานด้วยสายคำสั่ง เขียนขึ้นใน C และออกสู่สาธารณะภายใต้ GPL มักใช้ในการ mirror เว็บและ FTP sites (Web crawling, Retrieved 2012, January23)

2.3.5.5 GRUB เป็น Crawler ค้นหาที่กระจายทั่วไปและเป็นopen source ซึ่ง Wikia Searchใช้ในการค้นหาเว็บไซต์ (Web crawling, Retrieved 2012, January23)

2.3.5.6 Heritrix เป็น Crawler ที่มีคุณภาพในการเก็บข้อมูลของ Internet Archive ถูกออกแบบมาเพื่อให้เก็บข้อมูลบางส่วนของเว็บไซต์ต่างๆ ซึ่งมีจำนวนมาก เขียนขึ้นใน Java (Web crawling, Retrieved 2012, January23)

2.3.5.7 ht://Dig นั้นรวมเอา Web Crawler ไว้ในเครื่องมือการจัดทำดรรชนีincludes a Web crawler in its indexing engine. (Web crawling, Retrieved 2012, January23)

2.3.5.8 HTTrack ใช้ Web Crawler ในการสร้าง mirror ของเว็บไซต์สำหรับการเข้ามาดูแบบ off-line เขียนขึ้นใน C และออกสู่สาธารณะภายใต้ GPL (Web crawling, Retrieved 2012, January23)

2.3.5.9 ICDL Crawler เป็น Web Crawler ข้ามแพลตฟอร์ม ที่เขียนขึ้นใน C++ และตั้งใจที่จะ Crawl เว็บไซต์ต่างๆบนพื้นฐานของ Web-site Parse Templates โดยใช้ทรัพยากร CPU ที่ว่าง ของคอมพิวเตอร์เท่านั้น (Web crawling, Retrieved 2012, January23)

2.3.5.10 mnoGoSearch เป็น Crawler เป็นตัวจัดทำดรรชนี และเป็นเครื่องมือค้นหาที่เขียนขึ้นใน C และจดลิขสิทธิ์ภายใต้ GPL (เครื่อง Linux เท่านั้น) (Web crawling, Retrieved 2012, January23)

2.3.5.11 Nutch เป็น Crawler เขียนขึ้นใน Java และออกภายใต้ลิขสิทธิ์ Apache สามารถนำไปใช้ร่วมกับแพคเกจการจัดทำดรรชนีข้อความ Luceneได้ (Web crawling, Retrieved 2012, January23)

2.3.5.12 Open Search Server เป็นเครื่องมือค้นหาและซอฟต์แวร์ Web Crawler ที่ออกภายใต้ GPL (Web crawling, Retrieved 2012, January23)

2.3.5.13 Pavuk เป็นเครื่องมือสายคำสั่งสำหรับ Web Mirror พร้อมด้วยตัวเลือก Crawler X11 GUI และออกมาภายใต้ GPL มันมีลักษณะขั้นสูงหลายอย่าง เมื่อเทียบกับ wgetและ httrackเช่น มีกฎการกรองและการสร้างไฟล์ที่แสดงออกมาอย่างสม่ำเสมอ เป็นต้น (Web crawling, Retrieved 2012, January23)

2.3.5.14 PHP-Crawler เป็น Crawler บนพื้นฐานของ PHP และ MySQL อย่างง่าย ที่ออกภายใต้ BSD ติดตั้งง่าย และเป็นที่นิยมสำหรับเว็บไซต์ที่ขับเคลื่อนโดย MySQL ตัวเล็กๆ ที่อยู่บน Host ร่วม(shared hosting) (Web crawling, Retrieved 2012, January23)

2.3.5.15 tkWWW Robot เป็น Crawler ที่อยู่บนพื้นฐานของWeb Browser tkWWW (จดลิขสิทธิ์ภายใต้ GPL) (Web crawling, Retrieved 2012, January23)

2.3.5.16 YaCy เป็นเครื่องมือค้นหากระจายไปทั่วซึ่งฟรี สร้างขึ้นบนหลักการเครือข่าย peer-to-peer (จดลิขสิทธิ์ภายใต้ GPL) (Web crawling, Retrieved 2012, January23)

2.3.5.17 Seeks เป็นเครื่องมือค้นหากระจายไปทั่วซึ่งฟรี (จดลิขสิทธิ์ภายใต้ Affero General Public License) (Web crawling, Retrieved 2012, January23)

2.3.6 การCrawling เว็บไซต์เชิงลึก

หน้าเว็บจำนวนมากนั้นอยู่ในกลุ่ม เว็บไซต์ลึกหรือเว็บไซต์ที่มองไม่เห็น หน้าเว็บเหล่านี้มักจะเข้าถึงได้โดยการส่งคำค้นหาไปที่ฐานข้อมูล และ crawler ธรรมดานั้นไม่สามารถที่จะหาหน้าเหล่านี้ได้หากไม่มีการเชื่อมโยง (ลิงค์) ที่ชี้ไปที่พวกมัน โปรโตคอล Sitemap ของ Google และ mod oaiนั้นตั้งใจที่จะอนุญาตให้มีการค้นหาทรัพยากรเว็บเชิงลึกเหล่านี้ (Web crawling, Retrieved 2012, January23)

การ Crawling เว็บเชิงลึกนั้นเพิ่มจำนวนการเชื่อมโยงเว็บต่างๆที่จะต้อง Crawl ขึ้นเป็นทวีคูณ บาง Crawler เพียงแค่นำเอาURL ที่หน้าตาแบบนี้ <a href="URL" ไปเท่านั้น ในบางกรณี เช่น Googlebotการ Web Crawling นั้นจะกระทำกับข้อความทั้งหมดที่มีอยู่ภายในเนื้อหา hypertext, tags หรือ text (ข้อความ) (Web crawling, Retrieved 2012, January23)

2.3.7 การนำการCrawling Web 2.0 ไปใช้

2.3.7.1 Sheeraj Shah แสดงให้เห็นข้อมูลเชิงลึกของการนำการ Crawling Web 2.0 ที่ขับเคลื่อนโดย Ajax ไปใช้งาน (Web crawling, Retrieved 2012, January23)

2.3.7.2 ผู้อ่านที่สนใจอาจจะต้องการอ่าน AJAXSearch: Crawling, Indexing and Searching Web 2.0 Applications (Web crawling, Retrieved 2012, January23)

2.3.7.3 Making AJAX Applications Crawlable,จาก Google Code เอกสารนี้ให้คำจำกัดความข้อตกลงระหว่าง web servers และ crawlers เครื่องมือค้นหา ที่อนุญาตให้เนื้อหาที่มีการสร้างอย่างต่อเนื่องนั้นมองเห็นได้โดย Crawler โดย Google ก็สนับสนุนข้อตกลงนี้อยู่ในปัจจุบัน (Web crawling, Retrieved 2012, January23)

2.4 การจัดทำดรรชนีของเครื่องมือค้นหา

การจัดทำดรรชนีของเครื่องมือค้นหานั้น รวบรวม วิเคราะห์ และเก็บข้อมูลเพื่อช่วยให้เกิดการดึงข้อมูลอย่างรวดเร็วและแม่นยำ การออกแบบดรรชนีนั้น รวมเอาหลักการจากหลายๆ ศาสตร์เข้าด้วยกัน ทั้งภาษาศาสตร์ จิตวิทยาการรู้การเข้าใจ คณิตศาสตร์ สนเทศศาสตร์ ฟิสิกส์ และวิทยาศาสตร์คอมพิวเตอร์ ชื่อที่เวียนเปลี่ยนกันไปสำหรับกระบวนการ ในบริบทของเครื่องมือค้นหาที่ได้รับการออกแบบมาเพื่อค้นหาหน้าเว็บบนอินเตอร์เน็ตนั้น คือ การจัดทำดรรชนีเว็บ (Web Indexing) (Search engine indexing, Retrieved 2012, January23)

เครื่องมือที่เป็นที่นิยมนั้นเน้นไปที่การจัดทำดรรชนีข้อความทั้งหมด ของเอกสารออนไลน์ และใช้ภาษาธรรมชาติ ประเภทของสื่อ อาทิ วิดีโอและเสียงและกราฟิกนั้นก็สามารถค้นหาได้เช่นกัน (Search engine indexing, Retrieved 2012, January23)

Meta Search Engine (เครื่องมือค้นหา Meta) นั้นจะใช้งานดรรชนีของการบริการอื่นๆ ซ้ำอีก และจะไม่เก็บดรรชนีท้องถิ่น (local index) ไว้ ในขณะที่เครื่องมือค้นหาบนพื้นฐานของ cache นั้นจะเก็บข้อมูลดรรชนีไว้ตลอด พร้อมกับ corpus ด้วย การบริการข้อความบางส่วน จะจำกัดความลึกที่จัดทำดรรชนี เพื่อลดขนาดดรรชนีลง ไม่เหมือนกับดรรชนีข้อความทั้งหมด การบริการที่มีขนาดใหญ่ขึ้น มักจะดำเนินการในห้วงเวลาที่ระบุไว้ก่อนหน้าแล้ว เนื่องจากระยะเวลาที่ต้องการและต้นทุนในการดำเนินการ ในขณะที่เครื่องมือค้นหาที่อยู่บนพื้นฐานของ agent (ตัวแทน) จะจัดทำดรรชนีแบบทันที (real-time) (Search engine indexing, Retrieved 2012, January23)

2.4.1 การจัดทำดรรชนี

วัตถุประสงค์ของการจัดเก็บดรรชนีก็เพื่อให้ความเร็วและการทำงานมีประสิทธิภาพสูงสุด ในการค้นหาเอกสารที่เกี่ยวข้องกับคำค้นหา หากไม่มีดรรชนี เครื่องมือค้นหาจะต้องค้นหาเอกสารทุกเอกสารใน คลังข้อมูล ซึ่งกินเวลาและพลังงานคอมพิวเตอร์มาก ตัวอย่างเช่น ดรรชนีของเอกสาร 10,000 เอกสารนั้นสามารถค้นหาได้ภายในเวลามิลลิวินาที แต่การสแกนหาคำทุกคำ ในเอกสารใหญ่ๆ 10,000 เอกสาร อาจใช้เวลานานหลายชั่วโมง ต้นทุนที่เก็บข้อมูลคอมพิวเตอร์เพิ่มเติมเพื่อเก็บรักษาดรรชนี เช่นเดียวกับเวลาที่เพิ่มมากขึ้นเพื่อให้มีการปรับข้อมูล (อัพเดท) ข้อมูลให้เป็นปัจจุบัน สิ่งเหล่านี้ เป็นสิ่งที่จำเป็นจะต้องแลกกับเวลาที่น้อยลงในระหว่างการดึงข้อมูล (Search engine indexing, Retrieved 2012, January23)

2.4.1.1 ปัจจัยในการออกแบบดรรชนี

ปัจจัยหลักๆ ในการออกแบบโครงสร้างเครื่องมือค้นหานั้นมีดังนี้:

1) ปัจจัยผสมผสาน

ข้อมูลเข้าไปอยู่ในดรรชนีได้อย่างไร หรือ คำต่างๆ หรือ ลักษณะของหัวข้อถูกเพิ่มเข้าไปในดรรชนีในระหว่างการเชื่อมโยงข้ามไปมาระหว่างคลังข้อความ และดรรชนีที่มีหลายดรรชนีนั้นสามารถทำงานอย่างอิสระ (asynchronously) ได้หรือไม่ ตัวจัดทำดรรชนีจะต้องตรวจสอบเสียก่อนว่ามันกำลังทำข้อมูลเนื้อหาเก่าให้ทันสมัยหรอว่ากำลังเพิ่มข้อมูลใหม่เข้าไป การเชื่อมโยงข้ามไปมา (traversal) นั้นมักจะสัมพันธ์กับนโยบายการรวบรวมข้อมูล การผสมผสานดรรชนีเครื่องมือค้นหานั้นมีหลักการคล้ายๆกับคำสั่งการผสมผสาน SQL และอัลกอริทึ่มการผสมผสานอื่นๆ (Search engine indexing, Retrieved 2012, January23)

2) วิธีการเก็บรักษาข้อมูล

วิธีการเก็บรักษาข้อมูลดรรชนี นั่นคือ ข้อมูลควรจะเป็นข้อมูลที่ถูกบีบอัดหรือถูกกรอง ขนาดของดรรชนี (Search engine indexing, Retrieved 2012, January23)

ดรรชนีต้องการหน่วยความจำคอมพิวเตอร์มากเท่าไรในการสนับสนุนความเร็วในการค้นหา (Search engine indexing, Retrieved 2012, January23)

ความสามารถในการค้นหาคำหนึ่งคำในดรรชนีผกผัน ความเร็วในการค้นหาข้อมูลในโครงสร้างข้อมูล เปรียบเทียบกับความเร็วที่สิ่งนั้นจะได้รับการปรับปรุงให้ทันสมัย หรือถูกลบออก เหล่านี้เป็นหัวใจสำคัญของวิทยาศาสตร์คอมพิวเตอร์ (Search engine indexing, Retrieved 2012, January23)

3) การดูแลรักษา

ดรรชนีจะได้รับการดูแลรักษาอย่างไร (Search engine indexing, Retrieved 2012, January23)

4) ความทนทานต่อการทำงานที่ผิดพลาด

ความสำคัญของการบริการที่จำเป็นจะต้องเชื่อถือได้ ปัญหาต่างๆ นั้นมีตั้งแต่การจัดการกับดรรชนีที่มีปัญหา การระบุว่าข้อมูลไม่ดี (Bad Data) สามารถจัดการได้เดี่ยวๆหรือไม่ การจัดการกับฮาร์ดแวร์ไม่ดี การทำพาร์ทิชั่น และรายการต่างๆ เช่น การทำพาร์ทิชั่นแบบ hash-based หรือ composite เช่นเดียวกับกระบวนการ replication (Search engine indexing, Retrieved 2012, January23)

2.4.1.2 โครงสร้างข้อมูลดรรชนี

โครงสร้างเครื่องมือค้นหานั้นมีความหลากหลายในวิธีการในการดำเนินการจัดทำดรรชนีและวิธีการในการเก็บรักษาดรรชนีให้เป็นไปตามปัจจัยการออกแบบทั้งหลาย (Search engine indexing, Retrieved 2012, January23) ประเภทของดรรชนีมีดังนี้

1) Suffix Tree

โครงสร้างเปรียบเทียบได้กับต้นไม้ สนับสนุนการค้นหาแบบ linear time (เวลาเส้นตรง) สร้างขึ้นโดยการเก็บส่วนเสริมท้าย (suffix) ของคำแต่ละคำ Suffix Tree นั้นเป็นประเภทหนึ่งของ Trie โดย Trieนั้นสนับสนุนการ hashing ที่ยืดออกได้ ซึ่งเป็นสิ่งสำคัญในการจัดทำดรรชนีของเครื่องมือค้นหา โครงสร้างนี้ใช้สำหรับการค้นหาแบบแผนในลำดับ DNA และการรวมกลุ่มกัน ข้อเสียเปรียบใหญ่ๆ ก็คือการเก็บคำใน tree อาจจะต้องการพื้นที่มากกว่าการเก็บคำเฉยๆ การนำเสนออีกแบบหนึ่ง อยู่ในรูป suffix array (การจัดเรียง ส่วนเสริมท้าย) ซึ่งว่ากันว่า ต้องการ

หน่วยความจำเสมือน (virtual memory)ที่น้อยกว่า และสนับสนุนการบีบอัดข้อมูลเช่น อัลกอริทึ่ม BWT (Search engine indexing, Retrieved 2012, January23)

2) ดรรชนีผกผัน

เก็บรายการการปรากฏขึ้นของเกณฑ์การค้นหาย่อยแต่ละเกณฑ์ โดยมาก มักจะอยู่ในรูปแบบตาราง hash หรือ binary tree (Search engine indexing, Retrieved 2012, January23)

ดรรชนีอ้างอิง

เก็บการอ้างอิงหรือการเชื่อมโยงหลายมิติ ระหว่างเอกสารต่างๆ เพื่อสนับสนุนการวิเคราะห์การอ้างอิง โดยจะเป็นไปตาม Bibilometrics (Search engine indexing, Retrieved 2012, January23)

3) ดรรชนี Ngram

เก็บลำดับของความยาวของข้อมูลเพื่อสนับสนุนการดึงข้อมูลหรือการค้นข้อวาม (Search engine indexing, Retrieved 2012, January23)

4) Document-term matrix

ใช้ในการวิเคราะห์แบบ latent semantic เก็บการปรากฏขึ้นของคำต่างๆในเอกสารในรูปแบบของตารางวิเคราะห์สองมิติ (two-dimensional sparse matrix) (Search engine indexing, Retrieved 2012, January23)

2.4.1.3 ความท้าทายในการทำงานคู่ขนาน

ความท้าทายหลักในการออกแบบเครื่องมือค้นหาคือการบริหารจัดการกระบวนการคอมพิวเตอร์ที่ต่อเนื่องกัน อาจจะมีโอกาสมากมายที่จะเกิดภาวะการแข่งขันและข้อผิดพลาดซึ่งสอดคล้องกัน ตัวอย่างเช่น เอกสารใหม่ได้ถูกเพิ่มเข้าไปในคลังและดรรชนีจะต้องได้รับการปรับปรุงให้ทันสมัย แต่ในขณะเดียวกันดรรชนีนั้นก็ยังต้องตอบสนองต่อคำค้นหาต่างๆที่ถูกป้อนเข้ามา นี่เป็นการชนกันของงานที่ทำงานแข่งกันสองงาน ลองคิดว่า ผู้เขียนเป็นผู้ผลิตข้อมูล และ web crawler เป็นลูกค้าของข้อมูลนี้ ที่ค้นหาข้อความ และนำมาเก็บไว้ใน cache (หรือคลัง-corpus) ดรรชนีก้าวหน้าคือลูกค้าของข้อมูลที่ผลิตขึ้นโดยคลัง และดรรชนีผกผันคือลูกค้าของข้อมูลที่ผลิตขึ้นจากดรรชนีก้าวหน้า สถานการณ์เช่นนี้มักจะถูกเรียกว่า โมเดลผู้ผลิต-ลูกค้า (producer-consumer model) ตัวจัดทำดรรชนีเป็นผู้ผลิตข้อมูลที่สามารถค้นหาได้ และผู้ใช้ก็คือลูกค้าที่ต้องการจะค้นหา ความท้าทายนั้นเพิ่มขึ้นเมื่อทำงานกับหน่วยเก็บข้อมูลและกระบวนการที่กระจายไปทั่วในความพยายามที่จะประมาณการณ์ด้วยข้อมูลที่ได้รับการจัดทำดรรชนีที่มีอยู่มากกว่า โครงสร้างของเครื่องมือค้นหาอาจจะต้องนำ การประมวลผลแบบกระจาย มาใช้ ซึ่งในการประมวลผลแบบนี้นั้น เครื่องมือค้นหาจะประกอบไปด้วยเครื่องมือทำงานหลายเครื่องทำงานด้วยกันอย่างพร้อมเพรียง

กัน นี่จะช่วยเพิ่มความเป็นไปได้ของการเกิดปัญหาความไม่สอดคล้องกัน และทำให้มีความยากลำบากมากขึ้นในการดูแลรักษาโครงสร้างที่ทำงานพร้อมกัน กระจาย และคู่ขนานกัน (Search engine indexing, Retrieved 2012, January23)

2.4.1.4 ดรรชนีผกผันต่างๆ

เครื่องมือค้นหาหลายเครื่องมือได้นำเอาดรรชนีผกผันเข้ามาใช้ด้วย เมื่อทำการประเมินคำค้นหา ให้สามารถระบุตำแหน่งของเอกสารที่มีคำต่างๆ ที่ต้องการค้นอยู่และจัดอันดับเอกสารเหล่านี้ตามเกณฑ์ความเกี่ยวข้อง เนื่องจากดรรชนีผกผันนี้จัดเก็บรายการเอกสารต่างๆที่มีคำแต่ละคำอยู่ เครื่องมือค้นหาสามารถใช้การเข้าถึงโดยตรง (direct access) ในการค้นหาเอกสารที่เกี่ยวข้องกับคำแต่ละคำที่ค้นหา เพื่อให้สามารถดึงเอกสารที่เกี่ยวข้องออกมาได้อย่างรวดเร็ว ต่อไปนี้คือตัวอย่างประกอบง่ายๆ ของดรรชนีผกผัน (Search engine indexing, Retrieved 2012, January23)

ตารางที่ 2.1 ดรรชนีผกผัน

คำ	**เอกสาร**
The	เอกสาร 1, เอกสาร3, เอกสาร 4, เอกสาร 5
Cow	เอกสาร 2, เอกสาร 3, เอกสาร 4
Says	เอกสาร 5
Moo	เอกสาร 7

ที่มา : Wikipedia (http://en.wikipedia.org/wiki/Index_(search_engine))

ดรรชนีนี้สามารถระบุได้เพียงแค่ว่า คำคำนั้นปรากฏอยู่ในเอกสารนั้นๆ หรือไม่ เนื่องจากว่ามันไม่ได้เก็บข้อมูลเกี่ยวกับความถี่และตำแหน่งของคำเอาไว้ จึงถือกันว่าเป็นดรรชนี Boolean ดรรชนีรูปแบบนี้จะระบุว่าเอกสารใดตรงกับคำค้นหาแต่จะไม่เรียงลำดับเอกสารที่ตรงกันให้ ในการออกแบบบางชิ้น ดรรชนีจะรวมเอาข้อมูลเพิ่มเติมเช่นความถี่ของคำแต่ละคำในเอกสารแต่ละชิ้น หรือตำแหน่งของคำในเอกสารแต่ละชิ้น ข้อมูลตำแหน่งช่วยให้อัลกอริที่มการค้นหาสามารถระบุความใกล้เคียงของคำ เพื่อสนับสนุนการค้นหาวลีต่างๆ ความถี่นั้นสามารถใช้เพื่อช่วยในการเรียงลำดับความเกี่ยวข้องของเอกสารกับคำค้นหา หัวข้อต่างๆเหล่านั้นเป็นหัวใจสำคัญในงานวิจัยในเรื่องของการดึงข้อมูล (information retrieval) (Search engine indexing, Retrieved 2012, January23)

ดรรชนีผกผันเป็นเมตริกซ์ (ตาราง) วิเคราะห์ เนื่องจากคำทุกคำนั้นไม่ได้ปรากฏอยู่ในเอกสารแต่ละชิ้น เพื่อลดหน่วยความจำของคอมพิวเตอร์ที่ต้องใช้ ดรรชนีจะถูกเก็บโดยวิธีที่แตกต่างจากการจัดเรียงสองมิติ ดรรชนีนี้จะคล้ายกับ term document matrices ที่ถูกใช้งานโดยการวิเคราะห์แบบ latent semantic ดรรชนีผกผันนั้นสามารถพิจารณาได้ว่าเป็นรูปแบบหนึ่งของตาราง hash ในบางกรณี ดรรชนีนั้นเป็นรูปแบบหนึ่งของ binary tree ซึ่งต้องการหน่วยความจำเพิ่มเติมแต่อาจย่นระยะเวลาการค้นหาลงได้ ในดรรชนีที่มีขนาดใหญ่กว่านี้ โครงสร้างนั้นจะอยู่ในรูปแบบของตารางการแจกแจง hash (distributed hash table) (Search engine indexing, Retrieved 2012, January23)

2.4.1.5 การรวมดรรชนี

ดรรชนีผกผันนั้นจะถูกเพิ่มผ่านการผสมผสานหรือสร้างใหม่ การสร้างใหม่นั้นใกล้เคียงกับการผสมผสานแต่ในขั้นแรกจะลบเนื้อหาของดรรชนีผกผันก่อน โครงสร้างอาจได้รับการออกแบบเพื่อสนับสนุนการจัดทำดรรชนีเพิ่มเติมขึ้นเรื่อย ๆ ส่วนการผสมผสานนั้นจะมีการระบุเอกสารที่จะมีการเพิ่มเติมหรือมีการปรับปรุงให้ทันสมัย และจะมีการวิเคราะห์เอกสารเหล่านั้นให้ได้ผลออกมาเป็นคำ เพื่อความแม่นยำทางเทคนิค การผสมผสานจะรวมเอาเอกสารที่เพิ่งจะผ่านการจัดทำดรรชนีเข้าไปด้วย โดยปกติแล้วก็จะอยู่ในหน่วยความจำเสมือน และมีcache ของดรรชนีอยู่ในฮาร์ดไดรฟ์คอมพิวเตอร์ หนึ่งหรือหลายเครื่อง (Search engine indexing, Retrieved 2012, January23)

หลังจากการวิเคราะห์ ตัวจัดทำดรรชนีจะเพิ่มเอกสารที่อ้างถึงแล้วเข้าไปในรายการเอกสารสำหรับคำต่างๆที่เหมาะสม ในเครื่องมือค้นหาที่ใหญ่กว่า กระบวนการค้นหาคำแต่ละคนในดรรชนีผกผัน (เพื่อรายงานว่ามีการปรากฏของคำนั้นในเอกสาร) อาจจะกินเวลามากเกินไป และทำให้กระบวนการนี้ มักจะถูกแบ่งออกเป็นสองส่วน ส่วนการพัฒนาของดรรชนีก้าวหน้าและกระบวนการที่แบ่งแยกเนื้อหาของดรรชนีก้าวหน้าออกไปเป็นดรรชนีผกผัน ดรรชนีผกผันได้ชื่อนี้มาเพราะว่าเป็นตัวผกผันของดรรชนีก้าวหน้า (Search engine indexing, Retrieved 2012, January23)

2.4.1.6 ดรรชนีก้าวหน้า

ดรรชนีก้าวหน้าจัดเก็บรายการคำต่างๆสำหรับเอกสารแต่ละชิ้น รูปแบบต่อไปนี้เป็นรูปแบบอย่างง่ายของดรรชนีก้าวหน้า:

ตารางที่ 2.2 ดรรชนีก้าวหน้า

เอกสาร	**คำ**
Document 1	the,cow,says,moo
Document 2	the,cat,and,the,hat
Document 3	the,dish,ran,away,with,the,spoon

ที่มา : Wikipedia (http://en.wikipedia.org/wiki/Index_(search_engine))

หลักการเบื้องหลังการพัฒนาดรรชนีก้าวหน้าก็คือ ในขณะที่เอกสารต่างๆกำลังผ่านการวิเคราะห์ น่าจะเป็นการดีกว่าหากมีการจัดเก็บคำต่างๆสำหรับเอกสารแต่ละชิ้นในทันที การวาดโครงร่าง (Delineation) จะช่วยในเรื่องของกระบวนการทำงานของระบบอย่างอิสระ ซึ่งจะเลี่ยงปัญหาคอขวดของการปรับปรุงข้อมูลดรรชนีให้ทันสมัยได้บ้าง ดรรชนีก้าวหน้านี้ถูกจัดระเบียบให้เปลี่ยนแปลงตัวเองไปเป็นดรรชนีผกผัน ดรรชนีก้าวหน้าจริงๆแล้วก็คือรายการของคู่ที่มีเอกสารและคำ เปรียบเทียบโดยยึดเอกสาร การเปลี่ยนจากดรรชนีก้าวหน้าเป็นดรรชนีผกผันนี้เป็นเพียงแค่การจัดคู่โดยใช้คำเท่านั้น ดังนั้น ดรรชนีผกผันจึงเป็นดรรชนีก้าวหน้าที่มีการจัดกลุ่มคำใหม่ (Search engine indexing, Retrieved 2012, January23)

2.4.1.7 การบีบอัด

การสร้างหรือการดูแลรักษาดรรชนีของเครื่องมือค้นหาขนาดใหญ่นั้นสะท้อนให้เห็นถึงความท้าทายใหญ่หลวงในเรื่องของการเก็บรักษาและการประมวลผลข้อมูล เครื่องมือค้นหามากมายใช้ประโยชน์จากรูปแบบหนึ่งของการบีบอัดเพื่อลดขนาดของดรรชนีบนดิสก์ลง (Search engine indexing, Retrieved 2012, January23) ลองพิจารณาสถาณการณ์ต่อไปนี้ ซึ่งเกี่ยวกับเครื่องมือค้นหาในข้อความเต็ม (full text) ในอินเตอร์เน็ต

1) จากการประมาณการณ์พบว่ามีหน้าเว็บที่แตกต่างกันถึง 2,000,000,000 หน้า ในปี ค.ศ. 2000 (Search engine indexing, Retrieved 2012, January23)

2) สมมติว่าแต่ละหน้าเว็บมีคำอยู่ 250 คำ (ตั้งอยู่บนสมมติฐานว่าหน้าเว็บนั้นเหมือนกับหน้าของหนังสือนิยาย) (Search engine indexing, Retrieved 2012, January23)

3) ใช้เนื้อที่ 8 บิต (หรือ 1 ไบต์) ในการเก็บอักขระหนึ่งตัว บางรหัสอักขระก็ใช้ 2 ไบต์ต่อหนึ่งอักขระ (Search engine indexing, Retrieved 2012, January23)

4) จำนวนอักขระเฉลี่ยในแต่ละคำบนหน้าเว็บ อาจจะประมาณการณ์ได้ว่าอยู่ที่ 5 ตัว (Search engine indexing, Retrieved 2012, January23)

5) คอมพิวเตอร์ส่วนบุคคลนั้น โดยเฉลี่ยแล้วจะมาพร้อมกับเนื้อที่ความจำ 100-250 กิกะไบต์ (Search engine indexing, Retrieved 2012, January23)

ในสถานการณ์เช่นนี้ ดรรชนีที่ไม่มีการบีบอัด (สมมติว่าเป็นดรรชนีที่ไม่มีการผสมผสาน และเป็นดรรชนีอย่างง่าย) สำหรับหน้าเว็บ 2 พันล้านหน้าจะต้องจัดเก็บคำทั้งหมด 5 แสนล้านคำ ในอัตรา 1 ไบต์ต่อหนึ่งอักขระ หรือ 5 ไบต์ต่อคำ นี่จะต้องใช้เนื้อที่ 2500 กิกะไบต์ในการจัดเก็บ ซึ่งมากกว่าเนื้อที่ในฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ส่วนบุคคลเฉลี่ยถึง 25 เครื่องรวมกัน ความต้องการเนื้อที่จัดเก็บนี้อาจจะเพิ่มมากขึ้นสำหรับโครงสร้างหน่วยความจำกระจายที่ทนทานต่อข้อผิดพลาด ดรรชนีนั้นสามารถลดขนาดลงไปได้ถึงเศษเสี้ยวของขนาดปกติ ขึ้นอยู่กับวิธีการบีบอัดที่เลือกใช้ สิ่งที่ต้องแลกมาก็คือเวลา และพลังงานในการดำเนินการ ที่จะต้องใช้ในการบีบอัด และขยายออก (Search engine indexing, Retrieved 2012, January23)

เป็นที่น่าสังเกตว่า การออกแบบเครื่องมือค้นหาขนาดใหญ่นั้นจะรวมเอาต้นทุนในการจัดเก็บเช่นเดียวกับต้นทุนค่าไฟฟ้าในการให้พลังงานที่จัดเก็บเข้าไปด้วย การบีบอัดจึงเป็นเรื่องของต้นทุน (Search engine indexing, Retrieved 2012, January23)

2.4.2 การวิเคราะห์เอกสาร

การวิเคราะห์เอกสารจะแยกส่วนประกอบ (คำ) ของเอกสารหรือสื่อในรูปแบบอื่นๆ ออกเพื่อการใส่ดรรชนีก้าวหน้าและดรรชนีผกผันเข้าไป คำต่างๆ ที่พบจะถูกเรียกว่า token และในบริบทของการจัดทำดรรชนีเครื่องมือค้นหาและการประมวลผลภาษาธรรมชาติ การวิเคราะห์นั้นจะถูกเรียกกันโดยทั่วไปว่า tokenization บางครั้งก็ยังเรียกว่า การเพิ่มความชัดเจนให้กับขอบเขตของคำ การติดป้าย การแบ่งข้อความออกเป็นส่วนๆ การวิเคราะห์เนื้อหา การวิเคราะห์ข้อความ การค้นข้อความ การสร้างความสอดคล้องกัน การแบ่งคำพูดออกเป็นส่วนๆ การ lexingหรือการวิเคราะห์คำ ศัพท์เช่น “การจัดทำดรรชนี” “การวิเคราะห์” “tokenization” นั้นถูกใช้แทนกันได้ หากใช้เป็นคำพูดอย่างไม่เป็นทางการในองค์กร (Search engine indexing, Retrieved 2012, January23)

การประมวลผลภาษาธรรมชาตินั้น ในปี ค.ศ.2006 นั้นเป็นศาสตร์แห่งการวิจัยอย่างต่อเนื่องและการปรับปรุงเทคโนโลยี Tokenization นั้นเสนอให้เห็นความท้าทายมากมายในการดึงข้อมูลที่จำเป็นออกมาจากเอกสารเพื่อการค้นหาคุณลักษณะสนับสนุน การ Tokenization สำหรับการจัดทำดรรชนีนั้น จะรวมเอาเทคโนโลยีที่หลากหลายเข้าไปด้วย ซึ่งการนำไปใช้นั้นมักจะถูกเก็บเป็นความลับขององค์กร (Search engine indexing, Retrieved 2012, January23)

2.4.2.1 ความท้าทายในการประมวลผลภาษาธรรมชาติ

1) ความคลุมเครือของขอบเขตของคำ

ผู้ที่ใช้ภาษาอังกฤษเป็นภาษาแม่นั้นอาจจะคิดว่าการ tokenization เป็นงานที่ง่าย แต่นี่จะไม่ง่ายเลยในกรณีของการออกแบบตัวจัดทำดรรชนีพหุภาษา ในรูปแบบดิจิตอล ข้อความภาษาอื่นๆ เช่นภาษาจีน ญี่ปุ่น หรืออารบิกนั้นสะท้อนให้เห็นความท้าทายที่เพิ่มมากขึ้นเนื่องจาก คำต่างๆ ไม่ได้ถูกคั่นโดยช่องว่างสีขาว เป้าหมายในการ tokenization คือการระบุคำต่างๆ ที่ผู้ใช้จะหา ระบบเหตุผลเฉพาะภาษานั้นจะถูกใช้เพื่อระบุขอบเขตของคำอย่างเหมาะสม ซึ่งมักเป็นเหตุผลในการออกแบบตัววิเคราะห์สำหรับทุกภาษาที่มีการสนับสนุน (หรือสำหรับกลุ่มของภาษาที่มีตัวบอกขอบเขตและวากยสัมพันธ์ คล้ายๆ กัน) (Search engine indexing, Retrieved 2012, January23)

เพื่อให้ความช่วยเหลือด้วยการจัดอันดับเอกสารที่ตรงกันอย่างเหมาะสมนั้น เครื่องมือค้นหาหลายตัวเก็บรวบรวมข้อมูลเพิ่มเติมเกี่ยวกับคำแต่ละคำเช่น ภาษา หรือ หมวดหมู่ของคำ (หน้าที่ของคำในประโยค) วิธีการเหล่านี้ขึ้นอยู่กับภาษาด้วย เนื่องจากมีวากยสัมพันธ์ที่แตกต่างกันไปในแต่ละภาษา เอกสารต่างๆ มักจะไม่ได้ระบุภาษาของเอกสารหรือบ่งบอกอย่างชัดเจนและถูกต้องนัก ในการ Tokenizing เอกสารนั้น เครื่องมือค้นหาบางเครื่องพยายามที่จะระบุภาษาของเอกสารโดยอัตโนมัติ (Search engine indexing, Retrieved 2012, January23)

2) รูปแบบของไฟล์ (format) ที่หลากหลาย

เพื่อให้สามารถระบุว่าส่วนใดของเอกสารที่มีอักขระอยู่ จะต้องมีการจัดการรูปแบบของไฟล์อย่างถูกต้อง เครื่องมือค้นหาที่สนับสนุนรูปแบบของไฟล์หลากหลาย จะต้องสามารถเปิดและเข้าถึงเอกสารได้อย่างถูกต้องและสามารถดำเนินการ tokenize อักขระต่างๆของเอกสารนั้น ความผิดพลาดที่เกิดขึ้นกับที่เก็บข้อมูล (Search engine indexing, Retrieved 2012, January23)

คุณภาพของข้อมูลภาษาธรรมชาตินั้นอาจจะไม่สมบูรณ์เสมอไป มีเอกสารจำนวนหนึ่งโดยเฉพาะอย่างยิ่งบนอินเตอร์เน็ต ที่ไม่เป็นไปตามระเบียบการจัดการไฟล์ที่เหมาะสม อักขระทวิภาค อาจจะถูกเข้ารหัสในหลายๆ ส่วนของเอกสาร หากไม่มีการระบุอักขระเหล่านี้ และหากไม่มีการดำเนินการอย่างเหมาะสม คุณภาพของดรรชนีหรือการทำงานของตัวจัดทำดรรชนี อาจจะแย่ลงได้ (Search engine indexing, Retrieved 2012, January23)

2.4.2.2 Tokenization

ไม่เหมือนกับมนุษย์ที่มีความรอบรู้ คอมพิวเตอร์ไม่เข้าใจโครงสร้างเอกสารภาษาธรรมชาติ และไม่สามารถรับรู้คำและประโยคต่างๆได้โดยอัตโนมัติ สำหรับคอมพิวเตอร์เอกสารก็คือลำดับของไบต์ เท่านั้น คอมพิวเตอร์ไม่ “รู้” ว่าที่ว่างนั้นแยกคำต่างๆ ออกจากกัน มนุษย์ต่างหากที่จะต้องตั้งค่า ตั้งโปรแกรมให้คอมพิวเตอร์ระบุว่าอะไรเป็นคำเดี่ยว หรือคำโดด ซึ่งจะเรียกว่า token โปรแกรมดังกล่าวมักจะเรียกกันทั่วไปว่า tokenizer หรือ parser หรือ lexer เครื่องมือค้นหาหลายตัว เช่นเดียวกับซอฟต์แวร์ประมวลผลภาษาธรรมชาติอื่นๆ จะผนึกเอาโปรแกรมเฉพาะทางสำหรับการวิเคราะห์ (parsing) เช่น YACC หรือ Lex เข้าไปด้วย (Search engine indexing, Retrieved 2012, January23)

ในระหว่างการ tokenization นั้น parser จะระบุลำดับอักขระที่บ่งบอกถึงคำและองค์ประกอบอื่นๆ เช่น เครื่องหมายวรรคตอน ที่จะแสดงด้วยรหัสตัวเลข บางรหัสก็เป็นอักขระที่ไม่มีบนเครื่องพิมพ์ Parser ยังสามารถระบุเอกลักษณ์เช่น ที่อยู่อีเมล์ (e-mail address) หมายเลขโทรศัพท์ และ URLs ได้ เมื่อระบุ token แต่ละตัวแล้ว คุณสมบัติหลายข้อก็จะถูกเก็บไว้ เช่น ตำแหน่งของ token (อักขระบน อักขระล่าง ผสม เหมาะสม) ภาษา หรือการเข้ารหัส หมวดหมู่ของคำ (หน้าที่ของคำในประโยคเช่น เป็นคำนาม หรือคำกริยา) ตำแหน่ง หมายเลขประโยค ตำแหน่งของประโยค ความยาว และหมายเลขบรรทัด (Search engine indexing, Retrieved 2012, January23)

2.4.2.3 การระบุภาษา

หากเครื่องมือค้นหาสนับสนุนภาษาหลายภาษา ขั้นตอนเริ่มต้นร่วมกันระหว่างการ tokenization คือการระบุภาษาของแต่ละเอกสาร ขั้นตอนต่อๆมาหลายขั้นตอนนั้นจะขึ้นอยู่กับแต่ละภาษา (เช่นการติดป้ายที่มาที่ไป (stemming) และหน้าที่ของคำในประโยค (part of speech)) การระบุภาษา นั้นเป็นกระบวนการที่โปรแกรมคอมพิวเตอร์พยายามจะระบุหรือจัดหมวดหมู่ภาษาของเอกสารโดยอัตโนมัติ ชื่ออื่นๆ สำหรับการระบุภาษานั้นได้แก่ การจัดหมวดหมู่ภาษา การวิเคราะห์ภาษา การระบุภาษา และการติดป้ายภาษา การระบุภาษาโดยอัตโนมัตินั้นเป็นหัวข้อที่กำลังวิจัยกันอยู่ในการประมวลผลภาษาธรรมชาติ การค้นหาว่าคำต่างๆ นั้นอยู่ในภาษาใด อาจจะต้องใช้ตารางการระบุภาษา (language recognition chart) (Search engine indexing, Retrieved 2012, January23)

2.4.2.4 การวิเคราะห์รูปแบบ (format)

หากเครื่องมือค้นหานั้นสนับสนุนรูปแบบเอกสารหลายรูปแบบ เอกสารก็จะต้องมีการเตรียมพร้อมสำหรับการ tokenization ความท้าทายก็คือ รูปแบบเอกสารต่างๆ มีข้อมูลรูปแบบที่เพิ่มขึ้นมาจากเนื้อหาข้อความ ตัวอย่างเช่น เอกสาร HTML ก็จะมีป้าย HTML ติด ซึ่งจะบ่งบอกชี้เฉพาะถึงข้อมูลรูปแบบ เช่น การเริ่มต้นบรรทัดใหม่ การเน้นคำ และขนาดอักษรหรือรูปแบบของอักษร

หากเครื่องมือค้นหาไม่สนใจข้อแตกต่างระหว่างเนื้อหาและข้อมูล “ส่วนที่เพิ่มเข้ามา” ข้อมูลที่เพิ่มเติมเข้ามาจะถูกรวมเข้าไปอยู่ในดรรชนีด้วย ทำให้ผลการค้นหาออกมาไม่ดี การวิเคราะห์รูปแบบคือการระบุและจัดการกับเนื้อหาเชิงรูปแบบที่ฝังอยู่ในเอกสารซึ่งควบคุมแนวทางที่เอกสารนั้นแสดงออกมาบนจอคอมพิวเตอร์หรือแปลออกมาโดยโปรแกรมซอฟต์แวร์ การวิเคราะห์รูปแบบนี้ยังถูกเรียกว่า การวิเคราะห์โครงสร้าง การ parsing รูปแบบ การดึงป้ายที่ติดออก การดึงรูปแบบออก การจัดข้อความให้เป็นมาตรฐาน การทำความสะอาดข้อความและการเตรียมข้อความ ความท้าทายของการวิเคราะห์รูปแบบนั้นถูกทำให้ยุ่งยากมากขึ้นโดยความซับซ้อนของรูปแบบไฟล์ที่หลากหลาย รูปแบบไฟล์บางประเภทนั้นมีข้อมูลที่เปิดเผยออกมาไม่มาก ในขณะที่ไฟล์ประเภทอื่นๆ นั้นมีข้อมูลให้ค้นหาได้ทั่วไป (Search engine indexing, Retrieved 2012, January23)

1) รูปแบบไฟล์ที่พบกันทั่วไป มีข้อมูลเยอะ และได้รับการสนับสนุนจากเครื่องมือค้นหาหลายตัวนั้น (Search engine indexing, Retrieved 2012, January23) มีดังนี้

- HTML
- ไฟล์ข้อความ ASCII (เอกสารข้อความที่ไม่มีรูปแบบเฉพาะเพื่อให้คอมพิวเตอร์สามารถอ่านได้)
- Adobe's Portable Document Format (PDF)
- PostScript (PS)
- LaTeX
- UseNetnetnews server formats
- XMLและสิ่งที่ต่อยอดมาเช่น RSS
- SGML
- Multimediameta data เช่นรูปแบบ ID3
- Microsoft Word
- Microsoft Excel
- Microsoft Powerpoint
- IBM Lotus Notes

2) รูปแบบไฟล์ที่บีบอัด

ตัวเลือกในการจัดการกับรูปแบบต่างๆ นั้นได้แก่การใช้เครื่องมือวิเคราะห์ที่มีให้หาซื้อได้ทั่วไป ที่ออกโดยองค์กรที่ได้พัฒนา ดูแล หรือเป็นเจ้าของรูปแบบนั้น และการเขียน parser (ตัววิเคราะห์) ขึ้นมาเอง (Search engine indexing, Retrieved 2012, January23)

เครื่องมือค้นหาบางตัวนั้นสนับสนุนการตรวจสอบไฟล์ที่ถูกเก็บอยู่ในรูปแบบที่บีบอัดหรือเข้ารหัส เมื่อต้องทำงานกับรูปแบบที่บีบอัดนั้น ตัวจัดทำดรรชนีจะต้องขยายเอกสารออกมาก่อน ขั้นตอนนี้อาจจะทำให้เกิดไฟล์หนึ่งไฟล์หรือมากกว่า ซึ่งแต่ละไฟล์ที่ออกมานั้นจะต้องได้รับการจัดทำดรรชนีแยกต่างหาก (Search engine indexing, Retrieved 2012, January23) รูปแบบไฟล์ที่บีบอัด ที่มีการสนับสนุนอยู่ทั่วไปได้แก่:

- ZIP - Zip archive file
- RAR - RoshalARchive File
- CAB - Microsoft Windows Cabinet File
- Gzip – ไฟล์ที่ถูกบีบอัดด้วย gzip
- BZIP – ไฟล์ที่ถูกบีบอัดด้วย bzip2
- Tape ARchive (TAR), Unix archive file, ไม่มีการบีบอัด (ด้วยตัวมันเอง)
- TAR.Z, TAR.GZ or TAR.BZ2 - Unix archive ถูกบีบอัดด้วย GZIP หรือ BZIP2

3) ตัวอย่างของการใช้รูปแบบเอกสารไปในทางที่ไม่ดี

การวิเคราะห์รูปแบบอาจมีวิธีการพัฒนาคุณภาพ เพื่อหลีกเลี่ยงการรวม “ข้อมูลแย่” เข้าไปในดรรชนีด้วย เนื้อหานั้นสามารถจัดการข้อมูลรูปแบบเพื่อให้รวมเอาเนื้อหาเพิ่มเติมเข้าไปด้วย (Search engine indexing, Retrieved 2012, January23) ตัวอย่างของการใช้รูปแบบเอกสารไปในทางที่ไม่ดี ในการ spamdexing นั้นได้แก่

- การรวมเอาคำเป็นร้อยๆ เป็นพันๆ คำ เข้าไปในส่วนที่ซ่อนจากจอคอมพิวเตอร์ แต่มองเห็นได้ด้วยตัวจัดทำดรรชนี โดยการใช้การจัดรูปแบบ (เช่น ป้าย “div” ซ่อนเร้นใน HTML ซึ่งอาจจะรวมเอาการใช้ CSS หรือ Javascriptในการทำการนี้ด้วย) (Search engine indexing, Retrieved 2012, January23)

- การตั้งค่าสีฟอนต์ของคำในตำแหน่งด้านหน้า ให้เหมือนกับสีพื้นหลัง ทำให้คำต่างๆ ถูกซ่อน ผู้ที่อ่านจากหน้าจอคอมพิวเตอร์มองไม่เห็น แต่ตัวจัดทำดรรชนีสามารถมองเห็นได้ (Search engine indexing, Retrieved 2012, January23)

2.4.2.5 การระบุหมวด/ส่วน/ภาค (section)

หากดัชนีเครื่องมือค้นหาเนื้อหานี้เป็นถ้าเป็นเนื้อหาปกติที่มีคุณภาพของดัชนีและคุณภาพการค้นหาอาจจะลดลงอันเนื่องมาจากเนื้อหาแบบผสมและความใกล้ชิดคำที่ไม่เหมาะสม สองปัญหาหลักที่จะตั้งข้อสังเกต (Web Search Engine, Retrieved 2012, January23)

เครื่องมือค้นหาบางตัวรวมเอาการระบุภาคส่วนเข้าไปด้วยซึ่งก็คือ การระบุภาคส่วนที่สำคัญๆของเอกสารก่อนที่จะดำเนินการ tokenization เอกสารทุกชิ้นในคลังนั้น ไม่ใช่ทุกชิ้นที่สามารถอ่านได้เหมือนกับหนังสือที่มีการเขียนมาอย่างดี มีการแบ่งเป็นบทและหน้าต่างๆ เอกสารหลายๆ ชิ้นบนเว็บเช่น จดหมายข่าว และรายงานองค์กรนั้น มีเนื้อหาที่จัดวางไม่ถูกต้องอยู่ และมีส่วนเพิ่มเติมที่ไม่มีเนื้อหาหลักๆ (ที่เกี่ยวข้องโดยตรงกับเนื้อหาของเอกสาร) เช่น บทความนี้มีเมนูด้านข้างที่เชื่อมต่อไปยังหน้าเว็บอื่นๆ รูปแบบไฟล์บางประเภทเช่น HTML หรือ PDF นั้นอนุญาตให้เนื้อหาถูกแสดงออกมาในรูปของคอลัมน์ได้ แม้ว่าเนื้อหาจะถูกแสดงออกมา หรือตั้งใจให้ออกมาในพื้นที่แบบต่างๆ แต่เนื้อหาเพิ่มเติมดิบอาจเก็บข้อมูลนี้ ตามลำดับ คำต่างๆที่แสดงออกมาตามลำดับในเนื้อหาในเนื้อหาจากแหล่งที่มาดิบจะถูกจัดดรรชนีตามลำดับ แม้ว่าประโยคและย่อหน้าเหล่านี้จะถูกจัดให้อยู่ในส่วนที่แตกต่างกันบนจอคอมพิวเตอร์ หากเครื่องมือค้นหาจัดทำดรรชนีของเนื้อหานี้เสมือนว่ามันเป็นเนื้อหาปกติ คุณภาพของดรรชนีและคุณภาพของการค้นหาอาจจะแย่ลง (Search engine indexing, Retrieved 2012, January23)

การวิเคราะห์ส่วนอาจจะต้องการให้เครื่องมือค้นหา ใช้งานการระบบตรรกะการ render ของเอกสารแต่ละชิ้น โดยเฉพาะอย่างยิ่ง เนื้อหาคร่าวๆของเอกสารจริง และจากนั้นจึงค่อยจัดทำดรรชนีจากเนื้อหาคร่าวๆ นั้นแทน ตัวอย่างเช่น เนื้อหาบางเนื้อหาบนอินเตอร์เน็ตนั้นได้รับการ render โดยใช้ Javascriptหากเครื่องมือค้นหาไม่ render หน้าและประเมิน Javascriptภายในหนึ่งหน้า เครื่องมือค้นหาจะไม่สามารถ “เห็น” เนื้อหานี้ ในทำนองเดียวกันและจะจัดทำดรรชนีเอกสารอย่างผิดวิธี เนื่องจากเครื่องมือค้นหาบางตัวไม่สนใจในเรื่องการ render ผู้ออกแบบหน้าเว็บหลายๆ คนจึงหลีกเลี่ยงที่จะแสดงเนื้อหาผ่าน Javascriptหรือใช้ป้าย Noscriptเพื่อให้มั่นใจว่าหน้าเว็บได้รับการจัดดรรชนี อย่างเหมาะสม ในขณะเดียวกัน ความจริงอันนี้สามารถนำไปใช้เพื่อให้ตัวจัดทำดรรชนี “มองเห็น” เนื้อหาที่แตกต่างจากสิ่งที่ ผู้อ่าน มองเห็น (Search engine indexing, Retrieved 2012, January23)

เนื่องจากเนื้อหาที่ผสมปนเปและความใกล้เคียงของคำที่ไม่เหมาะสม มีปัญหาหลักๆ สองปัญหาที่พบ นั่นก็คือ:

1) เนื้อหาที่อยู่คนละส่วนนั้นถูกจัดเสมือนว่าเกี่ยวข้องกัน ในดรรชนี แม้ว่าในความเป็นจริงแล้ว จะไม่เกี่ยวข้องกันเลย (Search engine indexing, Retrieved 2012, January23)

2) เนื้อหาจัดระบบ “ด้านข้าง” นั้นถูกรวมให้อยู่ในดรรชนีด้วย แต่เนื้อหาด้านข้างนั้นไม่มีส่วนที่เกี่ยวข้องกับเนื้อหาของเอกสาร และทำให้ดรรชนีนั้นเต็มไปด้วยตัวแทนเอกสารที่แย่ (Search engine indexing, Retrieved 2012, January23)

2.4.2.6 การจัดทำดรรชนี Meta Tag

เอกสารเฉพาะมักจะมีข้อมูล Meta ฝังอยู่ เช่น ผู้เขียน คำสำคัญ คำอธิบาย และภาษา สำหรับหน้า HTML ตัว meta tag จะมีคำสำคัญ ที่สามารถนำไปใส่ไว้ในดรรชนีได้ เทคโนโลยีเครื่องมือค้นหาอินเตอร์เน็ตในสมัยก่อนจะจัดทำดรรชนีเฉพาะคำสำคัญที่อยู่ใน meta tags สำหรับดรรชนีก้าวหน้า เอกสารตัวเต็มจะไม่ถูกวิเคราะห์ ในสมัยนั้น การจัดทำดรรชนีข้อความเต็มยังไม่แพร่หลาย ยังไม่มีแม้แต่ฮาร์ดแวร์ที่จะสามารถสนับสนุนเทคโนโลยีเช่นนั้นได้ การออกแบบภาษาเพิ่มเติม HTML นั้นเริ่มแรกเลยก็รวมการสนับสนุน meta tags เพื่อให้ง่ายและเหมาะสมต่อการจัดทำดรรชนี โดยไม่ต้องใช้การ tokenization (Search engine indexing, Retrieved 2012, January23)

เมื่ออินเตอร์เน็ตเติบโตขึ้นในช่วงทศวรรษที่ 1990s องค์กรแบบเดิม (brick-and-mortar corporations) ต่างก็ผันตัวเอง “ออนไลน์” กันหลายองค์กร และสร้างเว็บไซต์องค์กรขึ้นมา คำสำคัญที่ใช้ในการอธิบายหน้าเว็บ (ซึ่งหน้าเว็บหลายหน้าในสมัยนั้นก็คือหน้าเว็บขององค์กรที่คล้ายกับใบโฆษณาสินค้า) เปลี่ยนแปลงจากคำอธิบายธรรมดา เป็น คำสำคัญที่มุ่งเน้นไปที่การตลาด ออกแบบมาเพื่อส่งเสริมการขาย โดยการนำหน้าเว็บขึ้นไปอยู่ในอันดับต้นๆ ของการค้นหาสำหรับคำค้นหาบางคำ การที่คำสำคัญเหล่านี้ถูกกำหนดมาเป็นการเฉพาะเพื่อประโยชน์ส่วนบุคคลนั้นทำให้เกิด spamdexingซึ่งกลายมาเป็นตัวกระตุ้นให้เครื่องมือค้นหาหลายตัวหันมาใช้เทคโนโลยีการจัดทำดรรชนีแบบข้อความเต็ม ในช่วงปี 1990s ผู้ออกแบบเครื่องมือค้นหาและบริษัทต่างๆ ทำได้เพียงใส่ “คำสำคัญด้านการตลาด” หลายๆ คำลงในเนื้อหาของหน้าเว็บก่อนที่จะค่อยๆปล่อยข้อมูลที่น่าสนใจและมีประโยชน์ออกมา เนื่องจากการขัดผลประโยชน์กับเป้าหมายทางธุรกิจของการออกแบบเว็บไซต์ที่เน้นผู้ใช้ ซึ่ง “ไม่ค่อยมีการเปลี่ยนแปลง” สมการความคุ้มค่าชั่วอายุลูกค้าได้ถูกเปลี่ยนไปเพื่อการรวมเอาเนื้อหาที่มีประโยชน์มากขึ้นเข้าไปในเว็บไซต์ด้วยความหวังที่จะรักษาผู้เข้ามาเยี่ยมชมเอาไว้ ในทำนองเดียวกัน การจัดทำดรรชนีข้อความเต็ม เป็นสิ่งที่เน้นประโยชน์ และเพิ่มคุณภาพของผลการค้นหาของเครื่องมือค้นหา เนื่องจากมันเป็นการก้าวออกมาจาก การควบคุมการจัดวางผลการค้นหาของเครื่องมือค้นหาเพื่อประโยชน์ส่วนบุคคล ซึ่งก็ทำให้เกิดการพัฒนาการวิจัยเทคโนโลยีการจัดทำดรรชนีข้อความเต็ม (Search engine indexing, Retrieved 2012, January23)

ในการค้นหาบนเครื่องคอมพิวเตอร์นั้น ผลการค้นหามากมายมี meta tags เพื่อเปิดทางให้ผู้เขียนตั้งค่าตามสั่งได้ว่าจะให้เครื่องมือค้นหาจัดทำดรรชนีเนื้อหาจากไฟล์หลายๆไฟล์ได้ โดยเนื้อหาเหล่านั้นไม่จำเป็นจะต้องมาจากเนื้อหาของไฟล์ การค้นหาบนเครื่องคอมพิวเตอร์นั้นอยู่ภายใต้การควบคุมของผู้ใช้มากกว่า ในขณะที่เครื่องมือค้นหาในอินเตอร์เน็ตนั้นจะต้องเน้นไปที่การจัดทำดรรชนีข้อความเต็มมากกว่า (Search engine indexing, Retrieved 2012, January23)

2.5 คำค้นหาเว็บ

คำค้นหาเว็บ เป็นคำค้นหาที่ผู้ใช้กรอกเข้าไปในเครื่องมือค้นหาเว็บ เพื่อให้บรรลุความต้องการข้อมูลของเขา/เธอ คำค้นหาเว็บนั้นมีเอกลักษณ์ตรงที่คำเหล่านี้ไม่มีโครงสร้างและมักจะไม่ชัดเจน คำเหล่านี้แตกต่างกันไปตามมาตรฐานภาษาของคำค้นหาซึ่งควบคุมโดยกฎวากยสัมพันธ์เคร่งครัด (Web search query, Retrieved 2012, January23)

2.5.1 ประเภท

มีหมวดหมู่กว้างๆ 4 หมวด ที่ครอบคลุมคำค้นหาเว็บเกือบทั้งหมด เครื่องมือค้นหามักจะสนับสนุนคำค้นหาแบบที่สี่ ซึ่งมีการใช้ไม่ค่อยบ่อยนัก (Web search query, Retrieved 2012, January23)

2.5.1.1 คำค้นหาข้อมูล เป็นคำค้นหาที่ครอบคลุมหัวข้อกว้างๆ (เช่น *Colorado หรือ trucks*) อาจจะปรากฏผลการค้นหาหลายพันผลที่เกี่ยวข้อง (Web search query, Retrieved 2012, January23)

2.5.1.2 คำค้นหาทางไปสู่เว็บ เป็นคำค้นหาที่หาเว็บไซต์ หรือเว็บเพจหนึ่งๆ (เช่น *youtube*หรือ *delta air lines*) (Web search query, Retrieved 2012, January23)

2.5.1.3 คำค้นหาทางธุรกรรม เป็นคำค้นหาที่สะท้อนความต้องการของผู้ใช้ในการดำเนินการอย่างใดอย่างหนึ่ง เช่น ซื้อรถหรือดาวน์โหลด screen saver (Web search query, Retrieved 2012, January23)

2.5.1.4 คำค้นหาเพื่อการเชื่อมโยง เป็นคำค้นหาที่รายงานการเชื่อมโยงของ web graph ที่ได้รับการจัดทำดรรชนีแล้ว(เช่นการเชื่อมโยง (ลิงค์) ใดที่เชื่อมไปที่ URL นี้ และ มีหน้ากี่หน้าที่ได้รับการจัดทำดรรชนี จากชื่อโดเมนอันนี้) (Web search query, Retrieved 2012, January23)

2.5.2 ลักษณะต่างๆ

เครื่องมือค้นหาเว็บที่เป็นธุรกิจที่สุด จะไม่เปิดเผยบันทึกการค้นหา ดังนั้นจึงเป็นการยากที่จะได้มาซึ่งข้อมูลเกี่ยวกับว่าผู้ใช้ได้ค้นหาอะไรบ้างบนเว็บ อย่างไรก็ตาม งานศึกษาในปี ค.ศ.2001 นั้นได้วิเคราะห์คำค้นหาจากเครื่องมือค้นหา Excite และแสดงให้เห็นลักษณะที่น่าสนใจบางประการของการค้นหาบนเว็บ (Web search query, Retrieved 2012, January23)

2.5.2.1 ความยาวของคำค้นหาโดยเฉลี่ยอยู่ที่ 2.4 คำ (Web search query, Retrieved 2012, January23)

2.5.2.2 ประมาณครึ่งหนึ่งของผู้ใช้ ใส่คำค้นหาเดี่ยวๆ ในขณะที่เกือบหนึ่งในสามของผู้ใช้ใส่คำค้นหาที่เป็นเอกลักษณ์ สามคำค้น หรือมากกว่า (Web search query, Retrieved 2012, January23)

2.5.2.3 เกือบครึ่งหนึ่งของผู้ใช้ดูผลการค้นหาเพียงแค่หนึ่งหรือสองหน้าแรกเท่านั้น (10 ผลการค้นหาต่อหน้า) (Web search query, Retrieved 2012, January23)

2.5.2.4 น้อยกว่า 5% ของผู้ใช้ใช้งานการค้นหาขั้นสูง (สำหรับผู้ประกอบการ Boolean เช่น และ, หรือ, ไม่ (AND, OR, NOT)) (Web search query, Retrieved 2012, January23)

2.5.2.5 คำสี่อันดับแรกที่ใช้มากที่สุดคือ (ว่าง), *and, of, และ sex.* (Web search query, Retrieved 2012, January23)

การศึกษาบันทึกคำค้นหาของ Excite อันเดิมนั้นพบว่า 19% ของคำค้นหานั้น มีคำที่บ่งบอกถึงภูมิศาสตร์ (เช่นชื่อสถานที่ รหัสไปรษณีย์ ลักษณะทางภูมิศาสตร์ ฯลฯ) (Web search query, Retrieved 2012, January23)

ในงานศึกษาบันทึกคำค้นหาของ Yahoo ในปี ค.ศ.2005 พบว่า 33% ของคำค้นหาต่างๆ ที่มาจากผู้ใช้คนเดียวกันนั้นเป็นคำค้นหาซ้ำ และ 87% ผู้ใช้จะคลิกที่ผลการค้นหาเดิม นี่แสดงให้เห็นว่าผู้ใช้หลายคนใช้คำค้นหาซ้ำเดิมในการเข้าดูข้อมูลเดิม หรือค้นหาข้อมูลเดิมอีกครั้ง การวิเคราะห์นี้ได้รับการยืนยันโดยเรื่องราวในบล็อกของเครื่องมือค้นหา Bing ที่ระบุว่าราว 30% ของคำค้นหาเป็นคำค้นหาทางไปสู่เว็บ (Web search query, Retrieved 2012, January23)

นอกจากนี้ การวิจัยเพิ่มเติมได้แสดงให้เห็นว่าตารางแจกแจงความถี่คำค้นหานั้นเป็นไปตามกฎเลขยกกำลัง (power law) หรือ กราฟเส้นโค้งแจกแจงทางยาว นั่นก็คือ มีคำเป็นสัดส่วนไม่มากที่อยู่ในบันทึกคำค้นหาขนาดใหญ่ (เช่น มากกว่า 100 ล้านคำค้นหา) ถูกใช้บ่อยที่สุด ในขณะที่คำที่เหลือ ถูกใช้งานไม่บ่อยนัก ตัวอย่างนี้ ของหลักการ Pareto (หรือกฎ80-20) ทำให้เครื่องมือค้นหา นำวิธีการเพื่อความเหมาะสมที่สุด เช่นการจัดทำดรรชนี หรือการทำพาร์ทิชั่นฐานข้อมูล การ caching หรือการ ดึงล่วงหน้า (pre-fetching) มาใช้ (Web search query, Retrieved 2012, January23)

แต่ในงานศึกษาเมื่อไม่นานมานี้ ในปี ค.ศ.2011 มีการค้นพบว่าความยาวของคำค้นหาได้เพิ่มมากขึ้นอย่างสม่ำเสมอตามกาลเวลา และความยาวเฉลี่ยของคำค้นหาที่ไม่ใช่ภาษาอังกฤษก็เพิ่มขึ้นมากกว่าคำค้นหาภาษาอังกฤษ (Web search query, Retrieved 2012, January23)

2.5.3 คำค้นหาที่มีโครงสร้าง

ในเครื่องมือค้นหาที่สนับสนุนการประกอบการ Boolean และ การใช้วงเล็บนั้น วิธีการที่ใช้กันมาแต่เดิมโดยบรรณารักษ์นั้นอาจนำมาใช้ได้ ผู้ใช้ที่กำลังหาเอกสารที่ครอบคลุมหัวเรื่องหลายๆ หัวเรื่องหรือ มุมมองนั้นอาจจะต้องการที่จะอธิบายแต่ละหัวเรื่องหรือมุมมองด้วยการแยกคำบ่งบอกลักษณะออกจากกันเช่น ยานพาหนะ *หรือ* รถ *หรือ* รถยนต์ คำค้นหาที่บ่งบอกมุมมอง คือการรวมเอามุมมองต่างๆ เช่น (electronic OR computerized OR DRE) AND (voting OR elections OR election OR balloting OR electoral) จะมีความเป็นไปได้สูงที่จะพบเอกสารเกี่ยวกับ electronic

voting แม้ว่าเอกสารเหล่านั้นจะละคำว่า “electronic” และ “voting” หรือ ละไว้ทั้งสองคำ (Web search query, Retrieved 2012, January23)