

ลำดับการแสดงผลการค้นหาข้อมูล

กำหนดการแสดงผลเรียงตามความต้องการ

- เรียงข้อมูลตามการระบุเอกสารที่ต้องการให้แสดงก่อน (ranking)
- เรียงข้อมูลตามจำนวนคำในเอกสาร
- เรียงข้อมูลตามวันและเวลา
- เรียงข้อมูลตามชื่อเอกสาร
- เรียงข้อมูลตามขนาดของไฟล์

ในส่วนของการกำหนดลำดับก่อนหลังของผลลัพธ์ โดยปกติเรียงตาม hit count ซึ่งใช้อัลกอริทึม "Vector-Space" ในการคำนวณคะแนนในแต่ละเอกสาร ซึ่งขึ้นอยู่กับปรากฏของคำค้นหาในเอกสาร และจำนวนคำที่พบ

ข้อมูลเพิ่มเติมจากเอกสาร *dtSearch Text Retrieval Engine's Reference*

Sorting by hit count

If no term weighting, field weighting, or positional scoring is applied, relevance is determined by simply counting hits. Every word that matches the search request counts as one hit. Each word in a phrase counts separately, so "first class mail" would be three hits.

Automatic relevance scoring

For relevancy-ranked searches, dtSearch uses a "vector-space" algorithm to calculate a score for each document that takes into account the relative rarity of the search terms and their density in the retrieved file. Infrequent terms count more heavily than common terms, and N hits in a short document count more heavily than N hits in a long document.

An additional positional scoring mechanism increases the score when hits occur close to each other or close to the top of the file. With positional scoring, hits that are close together and at the top of the file count much more heavily than other hits.

In the dtSearch Engine API, two flags in SearchJob control these types of scoring: use `dtsSearchAutoTermWeight` to enable the vector-space scoring, and `dtsSearchPositionalScoring` to enable the positional scoring. The two options can be combined, and using both is recommended.

Scaling of relevance scores

The percentage shown in search results is, for each document, score for that document as a percentage of the highest-scoring document in the search results list. (The best-matching document always has a score of 100%.)