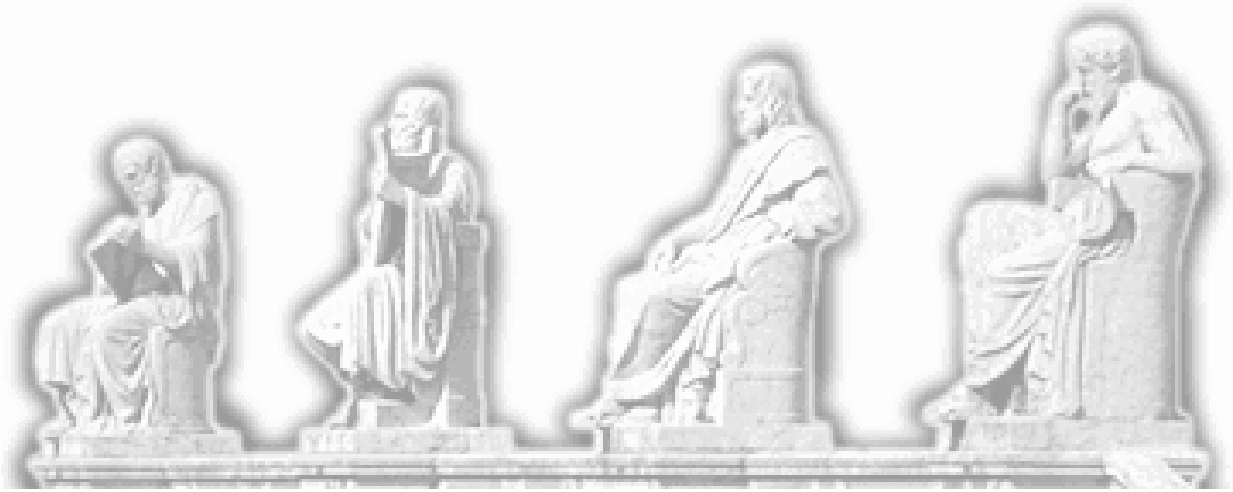
# Solr-Ranking – ein Mysterium oder Mission Impossible ?

Robert Scheuerl (BVB/A)



# Ausgangsituation

- Umstellung von webOPAC oder InfoGuide auf TouchPoint
  - Damit implizit Wechsel von FAST auf Solr
  - Solr ist die Standardsuchmaschine für TouchPoint, derzeit V4.4
  - Solr ist ein Lucene-Framework der Apache Software Foundation und open source
  - Elasticsearch ist ein anderes Lucene-Framework
- Wesentliche Solr-Funktionen
  - Facetten, Navigatoren, Drill-Down
  - Relevance Ranking
  - Performance

# Ausgangsituation

- Relevance Ranking vs. Sortierung
  - Ranking bedeutet Reihenfolge auf Basis eines Scorings zu definierten Sachverhalten bezogen auf Indexfelder eines Dokuments in Relation zur Sucheingabe
  - Sortierung bedeutet Reihenfolge auf Basis von festgelegten Algorithmen bezogen auf Felder eines Dokuments (z.B. Erscheinungsjahr oder alphabetisch)
- Implementierung in TouchPoint
  - Konfiguration der Standardsortierung
  - Unterschiedliche Algorithmen als bei FAST
  - Eindruck der Verschlechterung

# Ausgangsituation

- Ranking-Parameter
  - qf = free_search title^1 author^0.8 subject^0.8
    → Gewichtung der Index-Felder
  - pf = title^0.5 author^0.5 subject^0.5
    → Bewertung von Phrasen
  - bq = statusband:false^50
    → Zurückdrängen der Zeitschrifteneinzelbände
  - bf = recip(ms(NOW/YEAR,freshness),3.16e-11,1,1)^10.0
    → Bewertung der "Freshness" bzw. dem Erscheinungsjahr

Wichtig: Haben keinen Einfluss auf die Treffermenge !

# Ausgangsituation

- Standardalgorithmen
  - Scoring für Index-Felder (qf-Parameter)
    Standardmäßig zählt nur der höchste Score !
  - und für Phrasen (pf-Parameter)
    Die scores werden addiert !
  - Ermittlung der Scores nach TF/IDF (term frequency / inverse document frequency) Algorithmus
    Wie oft kommt ein Begriff oder eine Phrase im Dokument in einem Index-Feld vor und wie viele Begriffe gibt es im Dokument in Relation zur Größe des Index insgesamt
  - Zusätzlich Boosting für Nicht-Zeitschriftenbände
  - und “Freshness” bzw. Jahr

# Ausgangsituation

- Standardalgorithmen
  - qf- und pf-Parameter primär für die freie Suche
    d.h.: Suche ohne Angabe eines Indexfeldes !
  - Suche nach einem bestimmten Indexfeld
    Scoring reduziert sich auf dieses Indexfeld mit deutlich höherer Bewertung und Bewertung von Phrasen entfällt !
  - Suche nach Indexfeld, das nicht in qf-Parameter angegeben ist
    Kein Scoring auf Basis von Indexfeldern, nur Boosting wirkt, im wesentlichen also nach “Frehness”
  - Phrasen-Suche ist davon unabhängig
    Aber extrem hohe Bewertung !

# Ausgangsituation

- Standardalgorithmen
  - Sind diese für unsere Metadaten geeignet ?
    - Suchmaschinen arbeiten üblicher Weise mit Umfänglichen Daten mit wenig Struktur
    - Metadaten sind schlank mit viel Struktur

    → Nur bedingt geeignet
  - Experiment mit freasearch-Indexfeld
    - Schlanke Sätze wie Serien zu weit vorne
    - Umfängliche Sätze mit Kataloganreicherung zu weit hinten

    → Kompromiss muss gefunden werden

# Durchgeführte Änderungen

- Ranking-Parameter
  - qf = free_search title^1 author^0.8 subject^0.8

    neu:
  - qf = title^1 author^0.8 subject^0.3 institution^0.5 free_search^0.1
    - Titel und Autoren unverändert

      → sind meist die bestimmenden Faktoren
    - Schlagwörter abgewertet

      → waren zu bestimmend; evtl. ganz raus ?
    - Körperschaften dazu genommen

      → ähnliche Bewertung wie Autoren
    - Sammelsuchfeld extrem abgewerter

      → nur von geringer Relevanz

# Durchgeführte Änderungen

- Ranking-Parameter
  - pf = title^0.5 author^0.5 subject^0.5

    neu:
  - pf = title^0.04 author^0.1 institution^0.07
    - Deutliche Abwertung generell

      → die Scores sind sonst zu bestimmend
    - Schlagwörter raus

      → Einzelschlagwörter hier nicht sinnvoll; evtl. Ketten ?
    - Körperschaften dazu genommen

      → ähnliche Bewertung wie Autoren
  - Gute Balance erreicht – evtl. Autoren zurücknehmen ?

Wichtig: Parameter sind abhängig von Gesamtgröße des Index !

# Durchgeführte Änderungen

- Boosting-Parameter
  - bq = statusband:false^50

    → zurückdrängen der Zeitschrifteneinzelbände durch Boosting aller anderer Sätze, da kein negativ-Boosting
  - bf = recip(ms(NOW/YEAR,freshness),3.16e-11,1,1)^10.0

    neu:
  - bf = recip(ms(NOW/YEAR,freshness),3.16e-12,1,1)^30.0
    - Veränderte Formel und anderer Hebel für "Freshness"

      → Flachere Kurve

      → Höhere Bewertung der Freshness

# Ergebnis

- Aktuelle Situation
  - Besseres und nachvollziehbareres Ranking
  - Überlegung:

    Sollte man für das Ranking nur Informationen verwenden, die man in der Trefferliste sieht ?
- Weitere Parameter
  - tie = 0.2 (Wertebereich 0 bis 1)
    - Zusatzbewertung aller qf-Felder
  - pf2 = title^0,02 und ps2 = 2
    - Zusatzbewertung für Phrasen mit Wortabstand

# Weitergehende Überlegungen

- Zusätzliche Indexfelder für das Ranking
  - Titel-Indexfeld, das nur bestimmte Titelfelder umfasst, z.B. Haupttitel und Zusatz
  - Autoren-Indexfeld, dass die Autoren zusätzlich in umgekehrter Reihenfolge enthält

    Achtung: Scoring der Autoren hängt von der Anzahl der Autoren eines Titels ab !
  - Ermittlung der Freshness
    - Bei offenen Jahresangaben bei Zeitschriften aktuelles Jahr statt erstes verwenden ?
    - oder aktuellstes Jahr aus Bestandsangaben ?
    - „1000“ verwenden bei Titel ohne Jahr statt Aufnahmedatum ?

# Weitergehende Überlegungen

- Nutzergesteuert und dynamisch
  - Je nach Fachgebiert unterschiedliche Wünsche
  - Speicherung in den Benutzerdaten ?
  - Übergabe bei jeder einzelnen Suchanfrage

  Aber: großer Aufwand und Eigenentwicklung

# Vielen Dank für die Aufmerksamkeit

**Robert Scheuerl, Verbundzentrale**
089/28638-4253

robert.scheuerl@bsb-muenchen.de