Gestione dell'Informazione
Un sistema di full-text search per DBLP

Presentazione del progetto

DBLP

https://dblp.uni-trier.de

- Nasce come progetto dell'Università di Trier (Germania) con l'obiettivo di consentire l'accesso via web a dati bibliografici (ovvero pubblicazioni) nell'ambito della computer science…
- Oggi si definisce così (dalle F.A.Q.):

«The *dblp computer science bibliography* is the on-line reference for bibliographic information on major computer science publications. It has evolved from an early small experimental web server to a popular open-data service for the computer science community. Our mission at *dblp* is to support computer science researchers in their daily efforts by providing free access to high-quality bibliographic meta-data and links to the electronic editions of publications.»

- DBLP supporta ricerche full-text su tutti i campi indifferentemente oppure su un singolo campo (author, venue, pubblication)

DBLP XML

- Periodicamente viene effettuato un dump in XML dei dati bibliografici mantenuti da DBLP
- La cartella https://dblp.uni-trier.de/xml/ consente il download dei dump
- Nella root si trova sempre l'ultimo dump effettuato
- Il file dblp.dtd mostra lo schema del del document XML
- Nella cartella "release" si trovano una sequenza di dump rilasciati precedentemente
- Nella cartella «docu» trovate la documentazione e alcuni file java citati nei documenti

Il file dblp.xml

- Il file contiene innumerevoli informazioni bibliografiche (**elementi bibliografici** di seguito):
  - Articoli pubblicati su rivista (article)
  - Articoli pubblicati in atti di conferenze (inproceedings)
  - Capitoli di libri (incollection)
  - Atti di conferenze (proceedings)
  - Libri (book)
  - Tesi magistrali (mastersthesis)
  - Tesi di dottorato (masterthesis)
  - …
- Ogni elemento bibliografico ha una chiave univoca, valore dell'attributo `key`

Collegamenti fra elementi bibliografici

- Ogni elemento <inproceedings> e alcuni elementi <article> sono collegati al corrispondente elemento <proceedings> attraverso l'elemento <crossref>

```
<article mdate="2017-05-26"
key="journals/thipeac/VandeputteE11a">
<author>Frederik Vandeputte</author>
<author>Lieven Eeckhout</author>
….
<crossref>journals/thipeac/2011-4</crossref>
<url>db/journals/thipeac/thipeac4.html#Vandeput
teE11a</url>
</article>
```

```
<proceedings mdate="2017-05-26"
key="journals/thipeac/2011-4">
<editor>Per Stenstr&ouml;m</editor>
…
</proceedings>
```

- Lo stesso vale tra incollection e book
- E tra article e la corrispondente rivista?

Obiettivi del progetto

- Realizzare un sistema full-text search che consenta ad un utente di effettuare ricerche avanzate nella bibliografia di DBLP e che mostri i risultati ordinati in base a vari modelli di ranking (almeno 2)
- A tale scopo, il sistema dovrà occuparsi di due aspetti
  - La creazione e la gestione di un indice che contenga i dati bibliografici estratti da uno dei file di DBLP in XML
  - Il supporto alle ricerche full-text search sulla base del linguaggio mostrato nelle slide successive
    - Il sistema dovrà consentire all'atto della ricerca di selezionare il modello di ranking

Esempi di query

"information retrieval": ricerca la frase specificata in tutti gli elementi bibliografici di DBLP

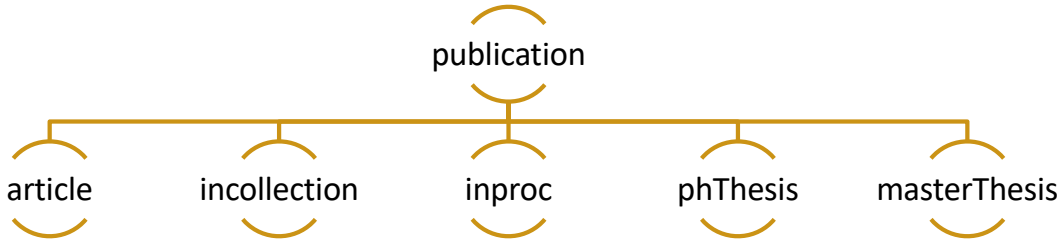
"information retrieval" VLDB: ricerca la frase specificata e la parola VLDB in tutti gli elementi bibliografici di DBLP

- Esempi di match sono
  - articoli pubblicati su conferenza il cui titolo contiene "information retrieval" e/o VLDB
  - Articoli il cui titolo contiene "information retrieval" e la conferenza contiene VLDB

publication.author: "Vianu" publication.year: 2018 venue: VLDB: ricerca tutte le pubblicazioni di Vianu pubblicate quest'anno in una rivista o conferenza che contiene l'acronimo VLDB

Sintassi del linguaggio per full-text search

- Il linguaggio si basa sulla seguente relazione di gerarchia fra i vari tipi di contributi



- La sintassi è la seguente:

f-t-s : ([field:] search-pattern)$^{+}$

search-pattern : keyword | “phase”

field: pub-search | *venue-search*

pub-search : pub-ele[.pub-field]

pub-ele: publication | article | incollection | inproc | phThesis | masterThesis

pub-field: author | title | year

venue-search: venue[.*venue-field*]

venue-field: title | publisher

Semantica del linguaggio per full-text search

- Una query del linguaggio ha la seguente forma:

  $search_1 \ldots search_n$ dove $search_i$=[$field_i$:] $search\text{-}pattern_i$

- La ricerca avviene sempre in OR (sulla logica dei modelli di ranking)
- Se $search_i$=$search\text{-}pattern_i$ la ricerca deve avvenire in tutti I campi di testo
  - Ad esempio "information retrieval" VLDB
- Se $search_i$=$field_i$:$search\text{-}pattern_i$ la ricerca deve essere realizzata nel campo specificato
  - *pub-search : pub-ele[.pub-field]*
    pub-ele: publication | article | incollection | inproc | phThesis | masterThesis
    pub-field: author | title | year

  facendo riferimento alla gerarchia, la ricerca deve essere limitata a pubblicazioni del tipo specificato o in tutte le pubblicazioni, nel caso di publication. Se il field non è specificato la ricerca avviene in tutti I campi di testo oppure limitatamente ai campi author o title o year.

  - *venue-search*: venue[.*venue-field*]
    venue-field: title | publisher

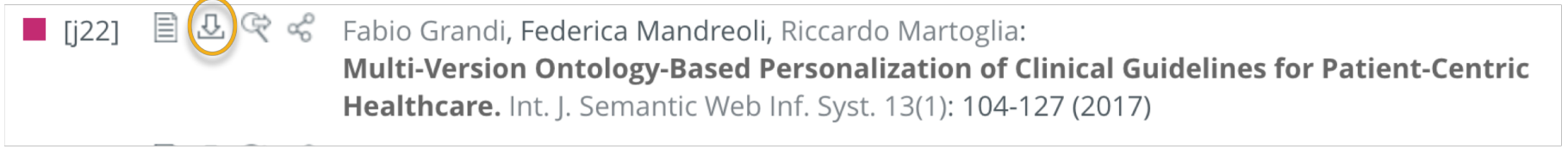
  la ricerca deve essere limitata ai venue ovvero inproceedings o book o journal. Se il field non è specificato la ricerca avviene in tutti I campi di testo oppure limitatamente ai campi publisher o title.

Alcune considerazioni sul meccanismo di matching

- Un **match** può essere
  - una pubblicazione
  - una venue
  - una pubblicazione e una venue
- Ad esempio data la query “big data analytics” VLDB, facilmente troveremo match con “big data analytics” nel titolo di pubblicazioni e VLDB nel titolo di atti di conferenza essendo VLDB l’acronimo di una conferenza di database
- Ad esempio la query article: “big data analytics” venue:VLDB richiede esplicitamente articoli che contengono la frase “big data analytics” pubblicati in riviste che contengono l’acronimo VLDB
- In caso di match che riguarda una pubblicazione e una venue è necessario combinare la pubblicazione con il relativo venue
- In questo caso anche i ranking andranno combinati
- Per la combinazione dei ranking si faccia riferimento al seguente lavoro che descrive un algoritmo noto come Threshold Algorithm (TA):
  R. Fagin, A. Lotem and M. Naor, Optimal aggregation algorithms for middleware, Journal of Computer and System Sciences 66, p. 614-656, 2003

Alcuni consigli…

- Prima di iniziare a progettare il sistema, cercate di comprendere appieno la “forma” dei dati
  - Navigare il sito delle pubblicazioni partendo ad esempio da un autore e accedendo alle sue pubblicazioni



  - Cliccando su export record è possibile vedere il corrispondente dato XML
  - Nel DTD trovate un link ad una pagina di DBLP che vi fornisce maggiori informazioni sulla struttura del file
  - Nella cartella ”doc” c’è un articolo scientifico che può aiutarvi in questa fase di analisi
- Quando progettate il sistema
  - È fondamentale progettare l’indice avendo in mente i vari tipi di query da supportare

Realizzazione e consegna del progetto

- Il progetto deve esser svolto in gruppo da 2 o 3 persone
- La valutazione del progetto sarà commisurata al numero di persone
- Al termina il gruppo dovrà produrre:
  1. un archivio (ZIP) contenente tutto il codice realizzato e un README per l'installazione e l'uso dell'applicazione
  2. una presentazione che descriva il sistema realizzato

1. Da consegnare una settimana prima dell'appello in cui verrà presentato il progetto
2. Da consegnare il giorno dell'appello

La presentazione

- Il numero di slide deve essere commisurato al tempo
- Nella presentazione devono essere mostrati I problemi tecnici affrontati e le soluzioni individuate
- Nel mostrare le soluzioni progettate è importante essere chiari su «come» il problema è stato risolto ovvero descrivere «quale» soluzione tecnica è stata individuata (approccio funzionale, metodologico) mentre non è necessario mostrare il codice, se non dei piccoli frammenti

Presentazione del progetto

- Il gruppo presenterà il progetto in occasione di un appello d'esame
  - Tempo 30 minuti (è molto importante rispettare i tempi)
  - Tutti i componenti del gruppo dovranno partecipare alla presentazione
- Il progetto deve essere presentato il giorno dell'esame orale o comunque prima dell'esame orale
  - È possibile presentare il progetto in un appello e fare l'orale in un appello successivo
- Solo per la **sessione di gennaio-febbraio 2018** sarà consentito di sostenere l'orale e poi successivamente presentare il progetto

L’esame…

- 60% del voto finale dipenderà dal voto dell’orale
  - 15 minuti di domande sugli argomenti del corso
- 40% del voto finale dipenderà dal voto del progetto e della presentazione
  - Il voto del progetto sarà personale
  - Il voto dipenderà dalla presentazione
  - Il voto dipenderà dalla qualità del sistema. Aspetti tecnici valutati:
    - Dimensione dei dati indicizzati
    - Progettazione dell’indice
    - Capacità dell’indice di supportare adeguatamente l’esecuzione di query
    - Algoritmi e soluzioni implementative adottate per l’esecuzione di query