Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea in Informatica per il management

Navigazione di articoli scientifici e citazioni: un ambiente scalabile e flessibile

Relatore:
Dr. Angelo Di Iorio

Presentata da:
Alice Graziosi

Sessione I
Anno Accademico 2015/2016

Sali sulle spalle dei giganti...

Introduzione

Questo lavoro di tesi si concentra sulle estensioni apportate all'applicazione BEX. BEX (Bibliographic Explorer) è una web app[1] finalizzata alla navigazione di pubblicazioni scientifiche attraverso le loro citazioni[1, 2]. Il settore in cui si colloca questo progetto è il *Semantic Publishing*, un recente ambito di ricerca che ha come scopo la pubblicazione sul web di articoli accademici a cui vengono associati metadati semantici. Le meta-informazioni che arricchiscono i documenti si riferiscono non solo al documento stesso, ma anche al suo contenuto. Gli *scholarly Linked Open Data* elaborati da BEX sono metadati strutturati secondo il modello RDF (Resource Description Framework)[2] e sono conformi alle moderne ontologie SPAR[3]. Il Semantic Publishing sfrutta tutte le tecnologie proprie del Semantic Web. Nel Semantic Web i contenuti disponibili online non sono solo semplici documenti HTML (o altro tipo), ma delle vere e proprie risorse corredate da metadati che provvedono a fornire al documento un contesto semantico (semantic context) in un formato adatto all'interrogazione e all'elaborazione automatica.

BEX nasce all'interno del Semantic Lancet Project[3] (SLP) [4, 5], un'iniziativa del Dipartimento di Informatica dell'Università di Bologna, il cui obiettivo è costruire un dataset aperto (Linked Open Dataset) di pubblicazioni accademiche, il Semantic Lancet Triplestore (SLT), e fornire strumenti per la navigazione ad alto livello e l'uso approfondito dei dati in esso contenuti. Originariamente BEX ha come backend il dataset Semantic Lancet Triplestore del Semantic Lancet Project, un Linked Open Dataset di statements

[1] Online in development mode su http://eelst.cs.unibo.it:8088/#/app/homeSearch, in produzione su http://eelst.cs.unibo.it:8089/#/app/homeSearch, su GitHub https://github.com/alicegraziosi/BEX

[2] https://www.w3.org/RDF/

[3] http://www.semanticlancet.eu/

RDF liberamente accessibile ospitato da uno SPARQL endpoint appositamente dedicato. Attualmente SLT contiene metadati relativi alle pubblicazioni del Journal of Web Semantics rilasciati dalla casa editrice Elsevier. BEX utilizza i metadati presenti nel dataset e ne offre all'utente visualizzazioni avanzate tramite un'interfaccia grafica ben ragionata, costituendo così un'applicazione interattiva e dotata di una buona user-experience. Questa applicazione permette in definitiva di navigare tra gli articoli accademici e esplorarli nel dettaglio dei loro riferimenti bibliografici. L'utente può conoscere i dati essenziali di un articolo (titolo, lista di autori, anno di pubblicazione, tipo di articolo, rivista o congresso di cui fa parte, DOI e abstract) e informazioni approfondite sulle citazioni in entrata, in uscita, sui contesti citazionali (*citation context*) e sulle funzioni citazionali (*citation function*).

L'utente di BEX è principalmente il ricercatore universitario che, per compiere le sue attività quotidiane, fa largo uso delle Digital Libraries (DL) e dei servizi che esse offrono. Alcune Digital Libraries fungono quasi solo da motori di ricerca di articoli accademici, mentre in altre più innovative sono stati compiuti ulteriori sforzi per l'elaborazione avanzata dei dati indicizzati. Dato il fermento dei ricercatori nell'ambito del Semantic Publishing è ragionevole pensare di ampliare e mantenere BEX in modo da offrire sempre nuovi servizi a seconda delle necessità che si presentano all'utilizzatore e per stare al passo con le funzionalità offerte dai sistemi esistenti allo stato dell'arte. La pubblicazione di scholarly Linked Open Data si sta velocemente diffondendo ed è per questo che è apprezzata l'estensione di un progetto che possa provvedere al sense-making di dati altrimenti interrogabili solo in modo diretto con query SPARQL. Inoltre non è comune che le grandi case editrici rilascino al pubblico i metadati come Linked Open Data in formato RDF, quando ciò accade è bene farne buon uso a favore del progresso della ricerca.

Considerando gli sviluppi futuri individuati per la versione precedente di BEX, le necessità degli utenti e ciò che offrono o meno i lavori correlati, sono state decise le novità da implementare. Il problema più importante era la dipendenza stretta dal dataset SLT: è necessario invece poter gestire altri datasets diversi in termini di volumi e struttura. Si è voluto inoltre aggregare i dati relativi al singolo autore e inserire viste *author-centric*

in modo da poter analizzare la produttività di un autore in relazione alle citazioni in entrata e uscita, anche in comparazione con un altro autore.

Le principali integrazioni apportate a BEX sono:

- l'**indipendenza** dal dataset SLT al fine di predisporre l'app all' uso di altri datasets dotati di strutture diverse;
- la **paginazione** dei risultati di ricerca e la scelta del numero massimo di risultati visibili per pagina per ovviare al problema della gestione di un numero potenzialmente alto di risultati ottenuti con una ricerca;
- l'aggiunta dei **dati aggregati** sul singolo autore, affinchè si abbiano viste *author-centric*;
- viste per la **comparazione** di due autori in base al numero di citazioni in entrata e in uscita, divise per anno e per tipologia;
- la ricerca in base alla venue di appartenenza degli articoli e la ricerca per nome completo dell'autore per raffinare la ricerca.

Si dedica la sezione 1.1 allo studio degli elementi da cui è composto il Semantic Web (RDF, SPARQL, Ontologie) e la sezione 1.2 alla descrizione del campo del Semantic Publishing. Anche i Linked Open Data sono un argomento molto in voga tra le comunità di ricerca in quanto il loro sfruttamento e riuso possono contribuire al progresso scientifico e alla diffusione della conoscenza e dunque se ne parlerà nel paragrafo 1.4. Si fa una rassegna delle Digital Libraries nel paragrafo 1.5, distinguendo le Digital Libraries Semantic Web-based (forniscono apertamente i metadati tramite Linked Open Data in formato RDF) dalle Digital Libraries non orientate al Semantic Web (non forniscono accesso diretto ai metadati tramite Linked Open Data in formato RDF). Maggiori spiegazioni sul Semantic Lancet Project si trovano nella sezione 2.2 e ulteriori informazioni su BEX nella sezione 2.3. Nel capitolo 3 sono introdotte le principali integrazioni apportate a BEX e la descrizione dei dettagli del contributo è trattata in modo più approfondito nel capitolo 4. Le conclusioni contengono infine sia le discussioni sulla valutazione, sui punti di forza e sui limiti di BEX, che gli sviluppi futuri pensati per ampliare il progetto.

Indice

Elenco delle figure

Capitolo 1

Semantic Web e Semantic Publishing

1.1 Semantic Web

BEX si inserisce nel campo di ricerca del Semantic Publishing che a sua volta è parte del contesto del Semantic Web, ciò che viene considerato l'evoluzione del World Wide Web tradizionale. Al termine Semantic Web, ideato da Tim Berners Lee nel 2001[6, 7], si associa l'idea di un ambiente dove agenti automatici riescano a elaborare e presentare agli utenti informazioni estrapolate dai contenuti. Nel Semantic Web si passa dal concetto di documento a quello di risorsa: ai documenti sono associati metadati che ne specificano il contesto semantico in un formato adatto all'interrogazione e all'elaborazione automatica. Le strutture dati prettamente *human-readable* tipiche del Web tradizionale vengono riformulate in modelli *machine-readable* adatti a essere processati automaticamente. Tramite le ontologie, di cui si parlerà in seguito, è possibile anche dare un significato alle semplici informazioni portando così l'informatica dalla computazione di dati verso un concetto di elaborazione automatizzata della conoscenza. Le annotazioni semantiche correlate a un documento riguardano sia il documento stesso che le entità presenti estratte dal testo, e vengono espresse sotto forma di statements RDF (Resource Description Framework)[1] machine-readable. Esistono motori semantici che estrapolano infatti informazioni anche

[1] https://www.w3.org/RDF/

dal parsing del testo di un documento ottenendo metadati a proposito delle entità e dei concetti contenuti in esso. I contenuti del Semantic Web non sono solo semplici documenti HTML (o di altro tipo) come nel web tradizionale, ma delle risorse più complesse e complete a ognuna delle quali è associato un URI (Uniform Resource Identifier) univoco sul web. I documenti sono così risorse interconnesse da link tipati (statements RDF) cioè da relazioni più avanzate rispetto ai semplici collegamenti ipertestuali (hyperlink). Scopo del Semantic Web è rendere machine-readable dati che altrimenti sarebbero solo human-readable, con l'arricchimento dei contenuti con meta-informazioni su cui si possono fare analisi, studi e statistiche.

1.1.1 RDF e SPARQL

RDF (Resource Description Framework)[2] è un modello per la codifica e lo scambio di metadati strutturati sul web. La rappresentazione dei dati in RDF è basata su triple soggetto-predicato-oggetto, uno statement RDF corrisponde a una tripla. Il soggetto di uno statement è sempre una risorsa (a cui è associato un URI), il predicato è una risorsa particolare usata per descrivere relazioni tra risorse e l'oggetto è un'altra risorsa o un valore letterale. Turtle è la sintassi più usata per la serializzazione dei dati RDF, insieme a RDFa, RDF/XML, N3, Turtle, e JSON-LD.

SPARQL (SPARQL Protocol and RDF Query Language)[3] è il linguaggio di interrogazione standard per queries su dataset RDF basato sul riconoscimento di un pattern su grafo. Esistono diversi SPARQL server, come Apache Jena Fuseki, che offrono REST API accettando queries SPARQL e restituendo risultati via HTTP. Uno SPARQL service è un triplestore cioè un sistema che ospita uno o più dataset che a loro volta possono contenere uno o più grafi RDF. Un grafo RDF corrisponde a una rete semantica ed è costituito da un insieme di statements RDF. È possibile fare SPARQL queries su un endpoint dataset alla volta, ma su più di un grafo contemporaneamente.

[2] https://www.w3.org/RDF/
[3] https://www.w3.org/TR/rdf-sparql-query/

1.1.2 Ontologie e OWL

Un'*ontologia* è un artefatto computazionale per modellare la struttura di un dominio, cioè per descrivere formalmente le entità e le relazioni che lo caratterizzano. É dunque una concettualizzazione progettata per esprimere il significato dei termini di un *vocabolario*. Esistono ontologie e famiglie di ontologie ideate specificatamente per i più vari domini: dalle risorse bibliografiche all'intelligenza artificiale, dall'ambiente giuridico-finanziario al campo medico.

OWL (Web Ontology Language) è l'attuale linguaggio di markup standard per definire, mediante l'uso di RDF, ontologie che permettono di modellare gli elementi di un dato sistema. OWL è un'estensione di RDF e la versione OWL 2 proviene da una recommendation[4] del W3C del 2012. Un sistema è composto da:

- **concetti**: (o anche classi o tipi) cioè entità ontologiche che possono essere istanziate, ognuna di esse rappresenta un gruppo di individui che condividono caratteristiche simili;
- **individui**: istanze di concetti, gli individui possono identificare oggetti concreti o astratti;
- **relazioni**: possono essere espresse tra individui o tra classi, sono predicati binari ognuno dei quali rappresenta un'entità ontologica che può riguardare coppie di classi o di individui.

1.2 Semantic Publishing

Recentemente l'uso delle tecnologie del Semantic Web applicate allo *Scholarly Publishing*[5] ha fatto nascere una nuova disciplina di ricerca: il *Semantic Publishing*. Con il termine Semantic Publishing ci si riferisce alla pubblicazione sul web di articoli accademici e risorse scientifiche arricchite da informazioni semantiche tramite l'uso delle

[4]https://www.w3.org/TR/owl2-overview/

[5]L'attività di pubblicazione di lavori accademici sotto forma di articoli su riviste specializzate, libri o tesi universitarie

tecnologie proprie del Semantic Web[8].

La distribuzione sul Web degli articoli scientifici può trarre beneficio dall'integrazione delle tecnologie del Semantic Web nei tool online che rendono accessibili questo tipo di risorse[9]. Il processo di *academic communication*[6] subisce singnificativi miglioramenti derivanti dal Semantic Publishing. La semplice ricerca di materiale sul web non basta, per una comprensione migliore dell'informazione c'è bisogno di strumenti che costruiscano significato intorno alle risorse e portino a una rappresentazione più ricca della conoscenza scientifica.

Le rivoluzioni nel Web e soprattutto l'evoluzione del Semantic Web hanno un impatto nella misura in cui vengono prodotti e condivisi sul web i risultati della ricerca scientifica[10]. Un limite del web tradizionale è il fatto che le informazioni sono solo *human-readable* cioè fruibili solo dalle persone. Uno dei goal del Semantic Web è strutturare le informazioni in modo che le macchine possano occuparsi anche della loro interpretazione oltre che della semplice esposizione. I tool per l'esplorazione di articoli scientifici pubblicati nel web hanno iniziato a includere *machine-readable markup* nei contenuti che ospitano. Da ciò scaturiscono migliori motori di ricerca e interrogazioni più precise da parte degli utenti.

Implementare le tecnologie del Semantic Web all'interno delle librerie digitali di articoli accademici portano all'ottimizzazione dell'uso di questo tipo di strumenti. La mera ricerca di titoli, autori e parole chiave all'interno di un portale per procurarsi informazioni utili porta a ottenere solamente risultati in linea con la ricerca in base alle occorrenze di parole trovate nei contenuti indicizzati. I risultati della ricerca non contengono informazioni correlate, collegamenti ad altre risorse e relazioni tra le entità presenti nel contenuto del documento. Quando invece un portale web fa uso delle tecnologie del Semantic Web[11] le ricerche portano a risultati ancora più consoni e a cui sono connesse molte informazioni di contorno di estrema utilità per l'utente. L'implementazione del Semantic Web nello sviluppo di Digital Libraries facilita l'utente nella ricerca, nell'accesso e nel recupero di informazioni rilevanti. I contenuti delle collezioni di risorse indicizzate dalle Digital Libraries dovrebbero essere annotate tramite ontologie che ne

[6]Pubblicazione e diffusione dei risultati della ricerca scientifica

possano esprimere dettagliatamente il significato e la descrizione semantica. L'applicazione delle ontologie alla modellazione delle informazioni rendono facilmente eseguibili task di elaborazione da parte di agenti automatici. L'integrazione di una DL con dati strutturati in modo tale che i contenuti siano marcati con un significato, porta alla creazione di un portale che offre non solo funzionalità di ricerca, ma anche servizi avanzati che hanno come conseguenza migliori modalità di condivisione della conoscenza e riuso delle risorse. I metadati che accompagnano le pubblicazioni fornendone un contesto semantico descrivono informazioni come il titolo, la lista di autori, l'anno di pubblicazione, la venue, il DOI, l'abstract e la bibliografia. Arricchire i documenti con i metadati facilita la ricerca automatica del documento stesso, e migliora il collegamento e la *data interation* con altri articoli della stessa o di sorgenti diverse. Grandi insiemi di metadati sono disponibili come LOD. I LOD (Linked Open Data) sono dataset aperti pubblicati sul web e liberamente accessibili dagli utenti a scopo di ricerca.

Pubblicare i risultati e i prodotti della ricerca tramite le tecniche del Semantic Web significa quindi corredare i documenti con meta-informazioni machine-readable. Il Semantic Publishing adduce indubbi benefici al sistema attraverso cui i lavori di ricerca e gli articoli accademici vengono creati, pubblicati, valutati, conservati, e diffusi all'interno della comunità scientifica. Contribuisce dunque a migliorare la *scholarly communication* non tanto sui canali formali come possono esserlo i *peer-reviewed journals*[7], ma soprattutto sui mezzi informali come i portali web delle Digital Libraries.

1.3 SPAR: Semantic Publishing and Referencing Ontologies

Le informazioni processate dall'applicazione BEX sono metadati RDF coerenti con le Semantic Publishing and Referencing Ontologies, altrimenti note come ontologie SPAR[8]. Le ontologie SPAR sono una suite di moduli ontologici basati su OWL 2 componibili e

[7]Selezione degli articoli da pubblicare su riviste specializzate fatta tramite valutazione di esperti nel settore

[8]http://www.sparontologies.net/

complementari appositamente introdotta per modellare nel dettaglio ogni aspetto del dominio del Semantic Publishing [12] e quindi descrivere in modo appropriato tutte le caratteristiche dei paper accademici e delle loro citazioni sottoforma di metadati machine-readable in formato RDF. Tramite SPAR vengono ampliate le ontologie progettate in precedenza per le pubblicazioni accademiche come DC Terms[9] e PRISM[10] colmandone le inefficienze e aggiungendo la possibilità di descrivere particolarità e informazioni più dettagliate. Le ontologie SPAR consentono la modellazione dei ruoli assunti dagli individui coinvolti nella pubblicazione (autore, editore, traduttore) e della variazione dello stato di un documento (sottoposto per pubblicazione, in stampa, in revisione, accettato per la pubblicazione) durante il processo di *submission-review-publication*. Questo insieme di vocabolari permette anche la caratterizzazione dei riferimenti bigliografici contenuti in un documento, prendendo in considerazione sia il contesto in cui sono state inserite sia il motivo (conferma, critica, estende, revisiona). Le ontologie SPAR più diffuse e utilizzate nel dataset analizzati con BEX sono:

- **FaBiO** (FRBR-aligned Bibliographic Ontology): un'ontologia fortemente basata su FRBR per descrivere genericamente articoli che contengono riferimenti bibliografici;
- **CiTO** (Citation Typing Ontology): un'ontologia con cui specificare il tipo di una citazione;
- BiRO (Bibliographic Reference Ontology): un'ontologia che permette di modellare collezioni di bibliografie;
- **C4O** (Citation Counting and Context Characterisation Ontology): un'ontologia costruita allo scopo di concettualizzare il numero di citazioni in entrata di un documento e il loro contesto citazionale(citation context) insieme al numero di citazioni globali in entrata contate fino a una certa data;
- **DoCO** (Document Components Ontology): un'ontologia per descrivere gli elementi della struttura di un articolo e le sue funzioni retoriche;

[9] http://dublincore.org/documents/dcmi-terms/

[10] Publishing Requirements for Industry Standard Metadata, http://www.idealliance.org/specifications/prism-metadata-initiative

– **DataCite** (DataCite Ontology): un'ontologia che implementa la specifica DataCite Metadata Schema[11] per la descrizione accurata delle proprietà delle bibliografie.

1.4 Linked Open Data

Da un'idea di Tim Berners-Lee[13, 14] nasce il concetto di Linked Data[12]: un insieme di dati strutturati interconnessi tra loro tramite *link tipati* cioè con statements RDF e quindi interrogabili per mezzo di queries semantiche. Dal momento che i Linked Data possono essere letti automaticamente dalle macchine è possibile interconnettere e interrogare dati provenienti da sorgenti differenti. Un contenitore di risorse viene anche chiamato *data silo*. I Linked Open Data (LOD) sono dataset contenenti dati interconnessi che vengono pubblicati sul web e resi accessibili sotto licenza libera. La pubblicazione dei LOD è incoraggiata dal progetto Linking Open Data[13] dell'organizzazione W3C che si propone di estendere il web impostando link RDF tra diversi data silo. Collegare tra loro archivi di Linked Open Data permette l'espansione e la navigazione del "Web of Data"[14]. Fondamentalmente la pubblicazione degli Open Data ha come nobile scopo la condivisione della conoscenza e il progresso scientifico. L'incentivazione a pubblicare i Linked Open Data ha avuto come conseguenza la nascita e la veloce espansione di un gigantesco grafo globale di dati composto da numerosi LOD interconnessi che è destinato a crescere ancora.

Il W3C ha esposto delle direttive[15] sulle modalità secondo cui i repositories di Linked Data debbano essere costruiti. Basato sui principi del Semantic Web, l'approccio consigliato prevede che a ogni entità rappresentata nei dataset sia associato un URI (Uniform Resource Identifier) HTTP e che venga descritta sotto forma di asserzioni in formato RDF con serialiazzione Turtle o simili. I dati accessibili nel web come Linked Open Data dovrebbero essere provvisti di uno SPARQL query service che ne permetta la libera interrogazione semantica.

[11] https://schema.datacite.org/
[12] http://linkeddata.org/
[13] https://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData
[14] https://www.w3.org/2001/sw/
[15] https://www.w3.org/DesignIssues/LinkedData.html

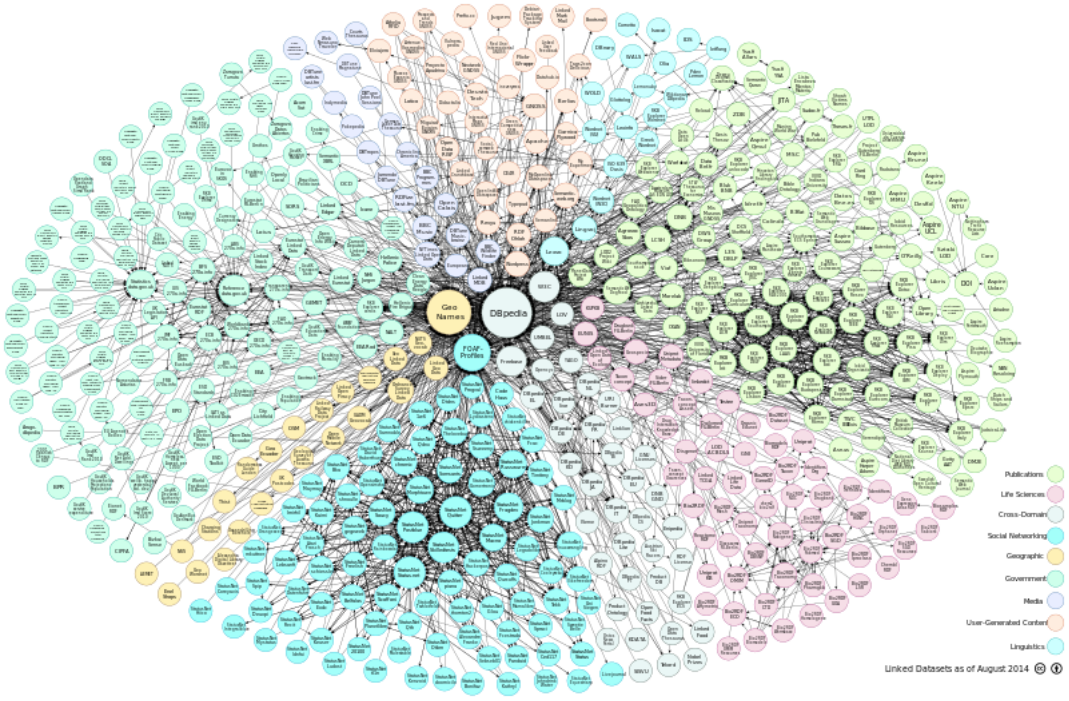

Figura 1.1: Interconnessione di Linking Open Data datasets, Agosto 2014. Fonte: http://lod-cloud.net/

In questa tesi ci si riferisce esclusivamente a Linked Open Data relativi al contesto delle pubblicazioni accademiche, ma in verità esistono Open Data sui più disparati settori del mondo reale. La comunità scientifica infatti attualmente opera, tramite lo sviluppo di piattaforme che sfruttano le tecnologie del Semantic Web e la progettazione di ontologie pertinenti, a favore della distribuzione e del *sense-making* di collezioni di dati che si riferiscono anche ad ambiti completamente diversi rispetto al mondo delle pubblicazioni accademiche, come per esempio insiemi di dati rilevati da sensori elettronici[15] chiamati *Linked Sensor Data*. Anche molte pubbliche amministrazioni e organizzazioni governative rendono apertamente accessibili i propri dati[16] soprattutto in un'ottica di trasparenza. Il Senato della Repubblica Italiana per esempio mette a disposizione un punto per l'accesso diretto a dati istituzionali costantemente aggiornati[17] su licenza

[16]https://www.w3.org/TR/gov-data/
[17]http://dati.senato.it/home

Creative Commons[18]. I dati, descritti formalmente con un'ontologia dedicata, possono essere interrogati eseguendo query su un endpoint SPARQL apposito[19]. In questo modo cittadini, giornalisti e ricercatori possono liberamente analizzare e riutilizzare i dati.

1.5 Strumenti del Semantic Publishing: le Digital Libraries

Dato il fermento e la sempre crescente attenzione da parte della comunità scientifica nei confronti del Semantic Publishing sono numerosi gli strumenti nati a supporto del ricercatore negli ultimi anni. Tra i risultati dell'applicazione delle tecnologie del Semantic Web al processo di pubblicazione ci sono i motori di ricerca semantici, le banche dati riguardanti il mondo accademico e universitario e le Digital Libraries (DL). I motori di ricerca semantici si distinguono da quelli tradizionali in quanto non fanno solo un matching delle occorrenze *full-text*, ma analizzano anche i metadati delle risorse indicizzate. Una Digital Library, o biblioteca elettronica, è un servizio online che si occupa di gestire e divulgare i prodotti della ricerca in formato digitale. Le Digital Libraries ospitano e rendono facilmente accessibili vaste collezioni di contenuti digitali allo scopo di diffondere i risultati degli studi scientifici e della ricerca. Le piattaforme di ricerca semantiche interpretano le informazioni associate ai documenti e permettono ricerche più sofisticate rispetto ai motori di ricerca classici che restituiscono come risultati dati non soltanto rilevanti, ma anche pertinenti. L'elaborazione dei metadati consente inoltre la formazione di una rete di collegamento tra i documenti secondo logiche più evolute.

Nonostante le entità che fanno parte processo di pubblicazione siano sempre state fonte di ispirazione nello sviluppo di standard del Semantic Web, non tutti i progetti di Digital Libary applicano pienamente le tecnologie del Semantic Web. Una delle prime tecnologie adottate è la specifica Dublin Core[20]. DCMI (Dublin Core Metadata Initiative) è un vocabolario creato da bibliotecari, editori, archivisti e accademici per catalogare risorse. Dublin Core permette di creare metadati di base per descrivere principalmente

[18]http://www.creativecommons.it/
[19]http://dati.senato.it/sparql
[20]http://dublincore.org/documents/dcmi-terms/

risorse bibliografiche, ed è probabilmente il vocabolario più utilizzato, ma risulta molto generico. DCMI è stato infatti ampliato da ontologie più precise come per esempio l'ontologia frbr[21] che implementa lo schema FRBR (Functional Requirements for Bibliographic Records)[22], FaBiO(FRBR-aligned Bibliographic Ontology)[23] appartenente a SPAR e fortemente fondata sulla specifica FRBR e ovviamente le altre ontologie SPAR di cui si è parlato in precedenza.

Generalmente i limiti maggiori a cui tutti gli sviluppatori di Digital Libaries si trovano davanti riguardano la scalabilità dei sistemi, la costruzione di interfacce utente accattivanti, la qualità dei metadati e l'aggiornamento dei datasets RDF. Non sempre le Digital Libaries si adoperano per stare al passo coi tempi e fornire servizi innovativi su dataset contenenti infomazioni aggiornate e recenti. Problemi di scalabilità compaiono quando si è in presenza di una grandissima quantità di dati da elaborare, ed è una difficoltà risolvibile usando sistemi di calcolo distribuito. La progettazione di interfacce grafiche sofisticate è assai rilevante per far sì che una DL diventi non soltanto popolare tra gli utenti, ma soprattutto di fondamentale utilità. Data la presenza di molteplici web tool simili tra loro è vantaggioso e indispensabile offrire all'utilizzatore un'interfaccia grafica che garantisca la semplicità d'uso di un'applicazione. Le Digital Libraries hanno sempre bisogno di essere aggiornate: alla pubblicazione di nuovi articoli bisogna integrare i metadati dei datasets preesistenti e rinnovare i riferimenti bibliografici in modo da assicurare la completezza dei dati. Un'ulteriore questione da gestire è la qualità dei metadati, può capitare infatti che ci siano dati duplicati o mancanti che generano problemi di disambiguazione, ripetizioni e inconsistenze. Infine, nonostante le DL spesso forniscano le informazioni sulle bibliografie, pochi sforzi sono compiuti per applicare sosfisticazioni grafiche su di esse, nel processo di *sense-making*, e nell'analisi avanzata.

È opportuno distinguere le Digital Libraries tradizionali da quelle orientate al Semantic Web. Sia le DL tradizionali che quelle non tradizionali, elaborano le annotazioni semantiche dei documenti dei loro database, ma le Digital Libraries Semantic Web-based

[21] http://vocab.org/frbr/core
[22] http://archive.ifla.org/archive/VII/s13/frbr/
[23] http://www.sparontologies.net/ontologies/fabio/source.html

si differenziano da quelle non Semantic Web-oriented in quanto, oltre a elaborarli, rilasciano anche i metadati relativi ai contenuti che ospitano, tramite triple RDF in un Linked Open Dataset a disposizione dell'utente. Di seguito si tratteranno i più popolari e significativi *search engine* di letteratura scientifica allo stato dell'arte, evidenziandone i lati positivi e le carenze.

1.5.1 Digital Library Semantic Web-oriented

Una Digital Library Semantic Web-based è una DL ricca di collegamenti interni ed esterni e risulta più completa rispetto a una DL tradizionale in quanto fornisce anche i metadati relativi ai suoi contenuti come LOD in formato RDF. Una semantic Digital Library integra informazioni derivate da diverse fonti di metadati (meta-informazioni sulle risorse, bookmarks e preferenze dei profili utente), garantisce l'interoperabilità con altri sistemi e fornisce un'interfaccia di browsing potenziata [16]. Le semantic DL incoraggiano la transizione da sistemi che offrono solo una fruizione statica delle informazioni a piattaforme collaborative[17, 18] e dinamiche in cui gli utenti sono anche contributori. Si parla anche di *Social Semantic Collaborative Filtering*: lo sfruttamento dei bookmarks e delle preferenze degli utenti allo scopo di condividere la conoscenza con una comunità di individui con interessi simili.

Scholarlydata

Il *conference Linked Open Dataset* di Scholarlydata[24] è la rivisitazione di quello di Semantic Web Dog Food (SWDF)[25] in un dataset aggiornato e completo[19, 20, 21]. Il modello dei dati usato per strutturare le triple RDF del dataset è la *conference-ontology*[26] con la quale si riformula l'ontologia preesistente *Semantic Web Conference Ontology* [27] grazie all'adozione di best practices nel design di ontologie. Il Linked Open Dataset è liberamente accessibile in diversi formati (HTML, RDF/XML, Turtle, N3, e JSON-LD) via URI dereferencing[28], tramite interrogazioni su SPARQL endpoint o scaricando singoli

[24]http://www.scholarlydata.org/
[25]http://data.semanticweb.org/
[26]http://www.essepuntato.it/lode/http://www.scholarlydata.org/ontology/conference-ontology.owl
[27]http://data.semanticweb.org/ns/swc/swc_2009-05-09.html
[28]Il recupero di una rappresentazione di una risorsa identificata da un URI

Figura 1.2: Risultato di una ricerca su DBLP++, una Digital Library Semantic Web-oriented

data dumps RDF per convegni e workshops.

Rexplore

È d'obbligo menzionare Rexplore[29]: un portale potente e innovativo che mette a disposizione dell'utente un ambiente interattivo per esplorare e rendere comprensibili i dati relativi alle pubblicazioni accademiche[22]. Rexplore usa tecniche di *visual analytics*[30] e permette, in un'interfaccia grafica avanzata, la visualizzazione di informazioni aggregate sulle pubblicazioni scientifiche. Rexplore è un progetto molto ampio e moderno, e in continua evoluzione: integra dati delle più importanti case editrici con quelli di risorse

[29] http://technologies.kmi.open.ac.uk/rexplore/

[30] I dati che esprimono un risultato vengono visualizzati graficamente in maniera interattiva

Figura 1.3: Metadati RDF forniti insieme al risultato di una ricerca su DBLP++

esterne come DBpedia[31], e il suo repository contiene metadati di più di 16 milioni di articoli. Rexplore offre servizi e visualizzazioni per:

- analizzare i trend e indentificare quali sono i topic più discussi e come variano nel tempo per tutte le aree della ricerca scientifica;
- scoprire argomenti emergenti all'interno della comunità di ricercatori;
- osservare come l'interesse degli autori cambiano da un argomento di ricerca a un altro e come avanza l'attività degli autori nel tempo;
- notare relazioni tra gli autori di pubblicazioni accademiche in base a quanto trattano argomenti simili o in base ai lavori svolti insieme;

[31]DPpedia è un progetto che ha come scopo l'estrazione di informazioni da Wikipedia e il loro rilascio sul web come Linked Open Data in formato RDF, http://wiki.dbpedia.org/

– analizzare dinamicamente le performance e di università, organizzazioni, nazioni, e gruppi di ricerca;

– notare le evoluzioni di gruppi di ricerca durante il tempo e le collaborazioni tra di essi;

– classificare automaticamente articoli, autori, congressi, e riviste specializzate in base al campo di ricerca.

DBLP++

Un sistema più di nicchia, ma non per questo meno interessante è DBLP++[32]: un portale la consultazione di bibliografie molto popolare all'interno della comunità di ricercatori dei dipartimenti di informatica. DBLP++ infatti offre accesso libero a una numerosa quantità di metadati riguardanti la bibliografia di articoli di ricerca inerenti a tematiche legate all'informatica. Il focus di DBLP++ è sullo stato dell'essere e la funzione un autore all'interno del panorama accademico. DBLP++ è stata la prima DL a esportare i propri dati in formato RDF ed è l'evoluzione di DBLP Computer Science Bibliography[33] che invece non era predisposta per mettere a disposizione i metadati.

Nature Publishing Group LOD Platform

Il Nature Publishing Group (NPG) è l'editore di Nature, la rivista più antica e prestigiosa nell'ambito della comunità scientifica internazionale. Il gruppo ha rilasciato i metadati, anche bibliografici, delle pubblicazioni comparse sulle loro riviste e li ha organizzati come Linked Open Dataset. Il LOD è provvisto di una piattaforma per la libera navigazione[34].

[32] http://dblp.l3s.de/dblp++.php, http://dblp.l3s.de/browse.php?browse=mostPopularKeywords
[33] http://dblp.uni-trier.de/
[34] http://data.nature.com

Springer LOD Conference Portal

Springer LOD Conference Portal[35] è un progetto nato allo scopo di rilasciare i metadati come Linked Open Data in formato RDF riguardanti *conference proceedings* pubblicati da Springer. L'intenzione degli sviluppatori del portale è quella di interconnettere il LOD dei metadati di Springer con gli altri dati disponibili nel LOD cloud.

1.5.2 Digital Library non Semantic Web-oriented

Esistono molte Digital Libraries incentrate sulle pubblicazioni accademiche che, pur analizzando i metadati semantici, non sono Semantic Web-based, ma su cui è interessante fare una panoramica in quanto molto popolari nel quadro d'insieme del mondo accademico.

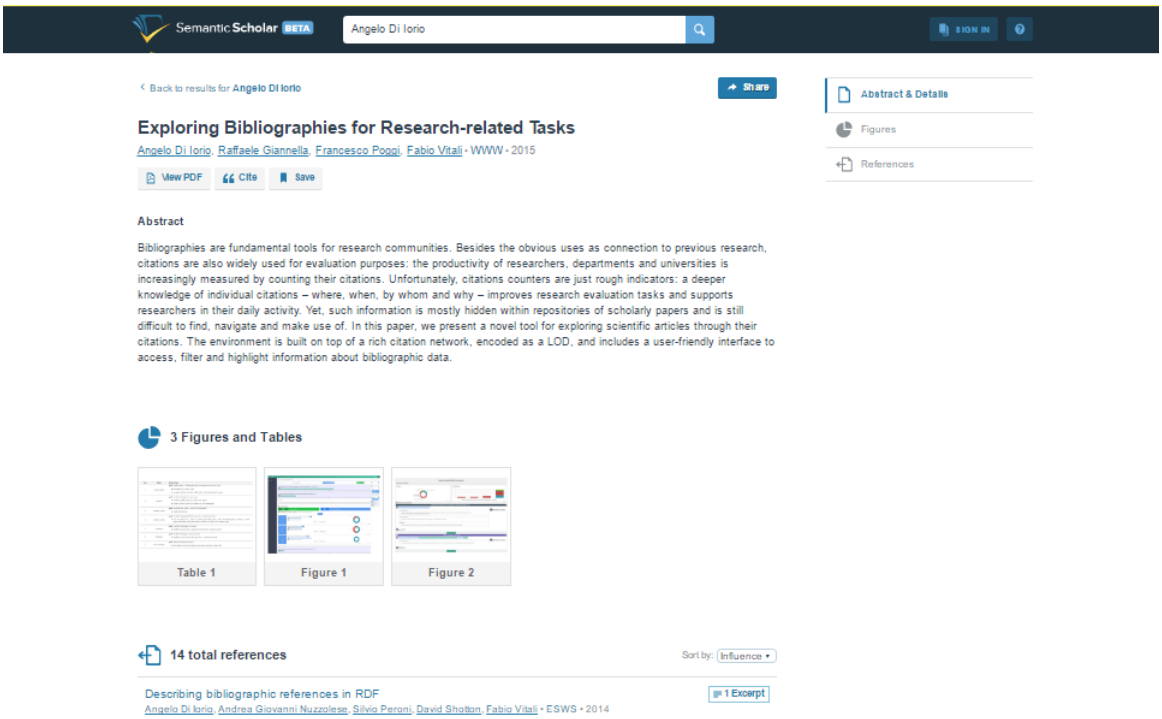

Figura 1.4: Risultato di una ricerca su Semantic Scholar, una Digital Library non Semantic Web-based

[35] http://lod.springer.com/data/search

CiteSeerX

CiteSeerX[36] è uno dei primi motori di ricerca di documenti accademici che siano stati sviluppati[23]. CiteSeerX si propone come scopo la diffusione e il miglioramento dell'efficienza nell'accesso alla conoscenza scientifica. Questo motore funge da Digital Library e indicizza più di 4 milioni di documenti riguardanti il campo dell'informatica e scienza dell'informazione. Il sistema ha tra le proprie caratteristiche l'estrazione automatica dei metadati dei contenuti che indicizza analizzando i contenuti del web, e il regolare aggiornamento delle citazioni degli articoli. CiteSeerX è una versione recente del sistema CiteSeer rilasciata dai suoi creatori per risolvere problemi di scalabilità e performance.

Semantic Scholar

Semantic Scholar[37] è un servizio nuovo e moderno progettato per portare in evidenza rapidamente ai ricercatori articoli rilevanti usando metodologie innovative di Intelligenza Artificiale[24]. Semantic Scholar è un progetto in espansione promosso dall'Allen Institute for Artificial Intelligence[38] il cui dataset include correntemente solo articoli di ricerca in informatica, ma che si prevede di estendere con articoli di altri ambiti scientifici. L'architettura sottostante a Semantic Scholar applica complessi algoritmi di *machine-learning* per analizzare gli articoli e estrapolarne i metadati essenziali con lo scopo di offrire al ricercatore papers qualitativamente interessanti in breve tempo garantendo un'eccellente user-experience. Il sistema estrae i .pdf dei papers dal web tramite crawler, li indicizza con Elasticsearch[39], ne estrae automaticamente metadati e citazioni e offre all'utente risultati di ricerca significativi e filtrabili con un'interfaccia grafica su misura per la ricerca accademica. Semantic Scholar si distingue per il fatto che le ricerche danno risultati rilevanti tenendo conto dell'influenza delle citazioni. La schermata principale infatti dà una prima idea generale di ciò su cui si concentra la piattaforma: vengono presentate delle statistiche sull'influsso degli autori, delle università e dei paper accademici calcolato esaminando il ranking delle citazioni. L'impatto di persone, istituzioni e articoli è

[36] http://citeseerx.ist.psu.edu/index

[37] https://www.semanticscholar.org/

[38] http://allenai.org/

[39] Elastic è il server di ricerca più utilizzato al mondo, è scalabile, distribuito e gestisce le informazioni in formato JSON, https://www.elastic.co/

valutato in base alle citazioni influenti (identificate quando l'articolo citato ha un impatto significativo sull'articolo citante) e alla velocità (numero medio di citazioni per anno durante gli ultimi tre anni). L'influenza delle citazioni è computata tramite metodi di intelligenza artificiale in base al numero e al contesto e aiuta il ricercatore a capire come un articolo si basa su un altro.

Scopus

Un ulteriore esempio è Scopus[40]: una Digital Library di articoli pubblicati su riviste scientifiche e libri o presenti nel programma di conferenze accademiche promosso dal gruppo editoriale Elsevier. Scopus è il database di abstract e citazioni più ampio in circolazione tra quelli sottoposti a sottoscrizione.

IEEE Xplore Digital Library

IEEE Xplore Digital Library[41] è una Digital Library di articoli accademici che espone i contenuti di una banca dati bibliografica delle pubblicazioni dell'Institute of Electrical and Electronics Engineers(IEEE). IEEE Xplore permette ricerche avanzate e filtri per contenuto, autore, titolo, anno ed editore.

Google Scholar

Il motore di ricerca di letteratura accademica più utilizzato e completo è probabilmente Google Scholar[42]. Google Scholar è un motore libero e indicizza il numero più vasto di documenti legati a tutte le aree della ricerca scientifica. La piattaforma permette di ricercare articoli rilevanti, esplorare i related works, le citazioni, gli autori, e ottenere il link al testo completo. Consente inoltre di creare un personale profilo utente come autore e controllare il numero di citazioni ricevute.

[40]https://www.elsevier.com/solutions/scopus
[41]http://ieeexplore.ieee.org/Xplore/home.jsp
[42]https://scholar.google.it/

Microsoft Academic

Un altro motore di ricerca aperto di articoli scientifici altrettanto diffuso è Microsoft Academic[43]. La rete semantica di Microsoft Academic contiene metadati riguardanti la bibliografia di papers pubblicati in riviste scientifiche o conference proceedings insieme a informazioni su riviste, congressi, autori e università. Diversamente da progetti simili offre presentazioni graficamente sofisticate dei dati e statistiche.

Academia.edu

Academia.edu[44] è una piattaforma di condivisione di papers accademici, destinata a professori e studenti universitari di tutti i settori, molto popolare anche tra persone non strettamente legate a progetti di ricerca. L'attenzione è concentrata sulla possibilità di caricare gratuitamente i propri articoli e condividerli online, magari con persone dagli interessi simili. Registrandosi e accedendo alla piattaforma si può fare una ricerca, scorrere il feed delle notizie, fare l'upload di documenti, segnare tra i preferiti gli articoli ritenuti interessanti, tenere traccia delle attività della propria lista di following, e monitorare le analisi sull'impatto che hanno i propri e gli altrui studi. Si sostiene che gli articoli pubblicati su Academia.edu ricevano, durante un periodo, di 5 anni una quantità maggiore di citazioni[25] rispetto a quante se ne otterrebbero con la sola pubblicazione su siti personali o dipartimentali.

ACM Digital Library

ACM DL[45] è una piattaforma di ricerca che offre l'accesso alla collezione full-text della maggior parte delle pubblicazioni appartenenti a riviste tecniche, libri e congressi dell'Association for Computing Machine[46]. Quello di ACM è un database bibliografico onnicomprensivo basato su argomenti relativi al Computing Machinery[47]. Compren-

[43] http://academic.research.microsoft.com/
[44] https://www.academia.edu/
[45] http://dl.acm.org/
[46] http://www.acm.org/
[47] Computazione automatica dei dati

de inoltre un ricco insieme di dati interconnessi rigurardo autori, papers, istituzioni e comunità di ricerca specializzate.

Capitolo 2

Semantic Lancet Project e BEX

2.1 Research-based tasks

Spesso non tutte le applicazioni esistenti hanno funzionalità complete: alcune elaborano in modo avanzato i dati del proprio database, altre invece forniscono solo servizi superficiali di ricerca. Una delle maggiori carenze è la mancanza di un interfaccia grafica che permetta la fruizione immediata e intuitiva dei contenuti da parte del ricercatore. Come anticipato, le Digital Libraries pubblicano i dati sugli articoli e sulle bibliografie in modo compatto e poco flessibile, e sfortunatamente sono applicate poche analisi raffinate sui dati. Sono infatti limitate le modalità in cui i contenuti dei database semantici ospitati dai portali vengono presentati al lettore, i portali offrono principalmente funzionalità di ricerca. Il ricercatore, mentre svolge le proprie attività di studio, consulta materiale interessante sulle Digital Libraries, ma ha anche bisogno di esplorare in modo approfondito i risultati della ricerca per esempio per verificare l'impatto che un articolo o un particolare autore ha avuto all'interno della comunità scientifica. I ricercatori impiegano molto tempo nella lettura degli articoli e delle loro bibliografie dato che generalmente la quantità di papers da spulciare è alta. Data la natura delle attività del ricercatore si rivela utile poter non solo ricercare documenti, ma anche esplorarli nel dettaglio delle loro bibliografie e per questi scopi non è dunque sufficiente la semplice ricerca di articoli offerta dalle piattaforme esistenti. Sistemi che possano semplificare e velocizzare la scelta di articoli rilevanti a cui fare rifermento per il proprio lavoro di ricerca sono

certamente apprezzati. Un aspetto interessante del lavoro del ricercatore è il modo in cui le bibliografie degli articoli che si consultano vengono sfruttate. La tendenza è quella di leggere un gran numero di articoli, ma di basarsi su un insieme limitato di articoli da citare. Questo capita perché ci si fonda principalmente su come le Digital Libraries mostrano la rilevanza di articoli di impatto, che a sua volta è dato da quanto un paper viene citato. Il ricercatore è al tempo stesso lettore, autore, *reviewer*, *evaluator* ed editore. In quanto lettore utilizza i riferimenti bibliografici per navigare tra gli articoli e trovarne di interessanti per il proprio ambito di ricerca. Anche nella scrittura di un proprio paper il ricercatore fa affidamento a ciò che trova di rilevante all'interno della miriade di documenti digitali. Spesso, all'interno di una comunità scientifica, articoli e proposte di progetto vengono sottoposte alla valutazione di esperti nel settore. Questo processo di *peer-review* prevede che chi se ne occupa controlli in profondità la bibliografia di un articolo, per trovare eventuali auto-citazioni e determinare quanto un articolo sia autoreferenziale o per vedere le innovazioni introdotte rispetto a lavori precedenti che vengono citati. Nella valutazione di un articolo il ricercatore ispeziona la distribuzione nel tempo delle citazioni ricevute e di quante citazioni vengono fatte ad altri articoli e in quale contesto. Valutando un autore si esaminano quante citazioni ha in entrata, il motivo per cui sono state fatte e come variano con gli anni. Come editore e organizzatore di congressi, l'accademico deve scegliere i ricercatori partecipanti e gli argomenti di cui discutere, anche in questo caso ha bisogno di navigare nel dettaglio dei riferimenti bibliografici. L'analisi dei riferimenti bibliografici è fortemente determinante per la valutazione di articoli e autori, e riveste dunque un ruolo fondamentale all'interno del lavoro del ricercatore. L'esistenza di un'applicazione che permetta di nagivare ad alto livello tra gli articoli attraverso le citazioni si rivela di significativo interesse.

2.2 Semantic Lancet Project

Originariamente il backend dell'applicazione BEX è il Semantic Lancet Triplestore (SLT), un ricco Linked Open Dataset di pubblicazioni scientifiche che fa parte dell'iniziativa Semantic Lancet Project[1] e interrogabile tramite le API REST di uno SPARQL

[1] http://www.semanticlancet.eu/

endpoint dedicato. Il Semantic Lancet Project è un gruppo di progetti complementari portato avanti dal Dipartimento di Informatica dell'Università di Bologna che si propone di creare e rendere liberamente accessibile un insieme di *rich scholarly data* (il Semantic Lancet Triplestore) organizzato in un Linked Open Data (LOD) e di fornire strumenti per sfruttarlo a pieno e analizzarne i dati in modo sofisticato.

Il framework del Semantic Lancet Project si compone di tre parti:

- **data reengineering**: il processo di traduzione dei *row data* provenienti da fonti esterne come Scopus[2] e ScienceDirect[3] in statements RDF conformi alle ontologie SPAR e loro integrazione in SLT allo scopo di sopperire a una condizione inconsistente dei dati;

- **semantic enhancement**: arricchimento del dataset SLT con altri dati semantici derivati dall'applicazione di script che estraggono metadati direttalmente dal testo degli articoli;

- **services**: tool ad alto livello complementari, ma non per forza dipendenti, realizzati per l'utente finale al fine di sfruttare i *semantically-enriched data* di SLT. Interrogare un dataset solo tramite l'interfaccia integrata di uno SPARQL endpoint con le REST API è fattibile solo per gli esperti del settore in quanto bisogna inevitabilmente conosere la struttura precisa con cui sono descritti i dati e le tecnologie del Semantic Web. Per facilitare la fruizione dei dati e allargare lo spettro dei possibili utenti è opportuno realizzare strumenti adatti per navigare i dataset e trarne informazioni utili. Di questi servizi fanno parte BEX, Data Browser[4], Abstract Finder [5], Citation Explorer[6] e Web Data Reporter[7].

[2]http://www.scopus.com
[3]http://sciencedirect.com
[4]http://www.semanticlancet.eu/browser/
[5]http://www.semanticlancet.eu/abstractfinder/
[6]http://www.semanticlancet.eu/citationexplorer/index.php
[7]http://www.semanticlancet.eu/reporter/index.html

2.2.1 Servizi offerti dal Semantic Lancet Project

Sviluppare applicazioni costruite *on top* rispetto al dataset RDF nasconde all'utente finale la complessità della struttura dei dati e i dettagli delle tecnologie che stanno alla base della loro interrogazione e elaborazione.

Data Browser è un tool che può essere comodo nella valutazione dell'attività di un certo autore: per ogni persona mostra gli articoli che ha scritto insieme ai dati di base come la lista degli autori, il titolo, informazioni sulla rivista tecnica dove il paper è stato pubblicato, l'abstract e il numero globale di citazioni in entrata.

Abstract Finder è una piattaforma di ricerca degli abstract di articoli che sfrutta le informazioni semantiche in formato RDF che riguardano le entità estratte del testo dell'abstract. I concetti che si possono identificare all'interno del testo sono persone, luoghi, organizzazioni, eventi, ruoli, temi, parole chiave e collegamenti a sorgenti esterne di open-data.

Citation Explorer è un servizio composto da tre moduli che serve per analizzare e costruire senso intorno alla rete dei riferimenti bibliografici. Nella prima sezione gli articoli del SLT sono divisi per anno e per ognuno è possibile vedere quali e quanti articoli cita o da cui è citato. La seconda sezione mostra, distinguibili per colore, i totali dei motivi citazionali *citation function* delle citazioni in entrata e in uscita di un dato articolo. Il terzo modulo, integrato nella nuova versione di BEX, mostra l'andamento del numero di citazioni in entrata e in uscita di due autori in comparazione, suddivise per anno e per quale ragione sono state fatte.

Web Data Reporter è un tool che permette di segnalare errori (per esempio multiple assegnazioni di DOI) del dataset, incompletezze (dati mancanti) e duplicazioni degli stessi dati in grafi diversi che portano potenzialmente a problemi di dereferenziazione e inconsistenza dei dati.

2.2.2 Il Semantic Lancet Triplestore

Attualmente il Semantic Lancet Triplestore contiene metadati relativi a tutti i papers pubblicati nel Journal of Web Semantics[8] di Elsevier per un totale di 367 articoli e circa 80000 statements RDF. L'idea però è quella di ripopolare il dataset con altri nuovi dati in futuro, ed è per questo motivo che l'architettura del progetto è organizzata in modo tale da poter gestire nuovi dataset provenienti da case editrici diverse appena essi diventino liberamente accessibili.

I *rich scholarly data* contenuti in SLT riguardano una vasta rete di citazioni tra articoli e dettagli sulle bibliografie e sono espressi in triple RDF conformi alla suite di ontologie SPAR. Il dataset è ospitato da un SPARQL query service[9] apposito che ne permette l'interrogazione. Il triplestore ad oggi in uso è Apache Jena Fuseki Server v2[10] che fornisce dati RDF tramite chiamate HTTP. Per ora è una piattaforma sufficiente, ma data la quantità potenzialmente enorme dei dati dei LOD e la limitatezza di questo mezzo, in futuro potrà essere necessaria la migrazione a un server più potente.

Le ontologie SPAR maggiormente usate nel dataset sono CiTO(Citation Typing Ontology), C4O (Citation Counting and Context Characterization Ontology) e DoCO (Document Components Ontology). Con queste ontologie sono modellati i dettagli relativi alle references: le citazioni globali in entrata e in uscita, i *citation context* (in quali frasi un paper cita un altro paper) e le *citation functions* (i motivi per cui un paper è stato citato). La principale novità introdotta allo stato dell'arte da parte di BEX è la possibilità di ricercare pubblicazioni accademiche e esplorarle attraverso la rete citazionale la quale è composta dalle citazioni in uscita (bibliografia) e quelle in entrata. Per ogni articolo è possibile conoscere il numero di citazioni globali ricevute, quante volte, in quale frase e perché un articolo cita un altro articolo o viene citato da un altro articolo. BEX fornisce inoltre informazioni aggregate sulle citazioni sotto forma di grafici e meccanismi avanzati per filtrare i risultati per esempio in base all'anno di pubblicazione o al tipo di articolo.

[8] http://www.journals.elsevier.com/journal-of-web-semantics/

[9] http://two.eelst.cs.unibo.it:8181/control-panel.tpl

[10] Un framework Java open source per creare applicazioni con le tecnologie del Semantic Web e i Linked Data, https://jena.apache.org/index.html

2.3 BEX: Bibliographic Explorer

Dall'interesse del gruppo di ricerca di Semantic Lancet Project di fornire sistemi per trarre informazioni utili dai dati di SLT ha origine BEX (Bibliographic Explorer)[11]: un'applicazione web interattiva dall'interfaccia user-friendly che ha lo scopo di facilitare l'esplorazione di articoli e dei loro riferimenti bibliografici. BEX nasce dalle esigenze del ricercatore che è al tempo stesso autore, lettore, reviewer ed editore di articoli accademici. Mediante BEX il ricercatore può infatti compiere tasks che abitualmente avrebbe portato avanti manualmente come per esempio controllare le autocitazioni di un articolo o filtrare la bibliografia. Dai limiti dei sistemi attualmente esistenti sono state progettate interfacce grafiche elaborate: il ricercatore ha bisogno di grafici esplicativi e manipolazioni dei dati per conoscere per esempio l'influenza di un certo autore, l'evoluzione di come viene trattato un argomento nel tempo, o l'impatto che ha avuto un autore o un progetto sulla comunità scientifica. Le citazioni, oltre a collegare per definizione un articolo a lavori precedenti, sono utilizzate per la valutazione di autori, università ed enti di ricerca in quanto la loro produttività e la loro popolarità è anche data dal conteggio delle citazioni riceveute. Un ulteriore goal che si intende raggiungere è il poter navigare attraverso le bibliografie e conoscerle approfonditamente in modo da sapere quale articolo cita un altro articolo, in quale frase e per quale motivo. Soddisfare questo proponimento significa migliorare la valutazione dei risultati della ricerca e aiutare il ricercatore nel suo lavoro. BEX è fondamentalmente un tool per la visualizzazione di una base dati di articoli e la navigazione di essi tramite le citazioni.

Il design del layout dell'applicazione è stato ben pensato e costruito in base al principio di *Information Seeking*[12] di Shneiderman "Overview first, zoom and filter, then details-on-demand"[26] che suggerisce che un'informazione vada prima presentata solo in anteprima insieme con la possibilità di applicare filtri e poi, se richiesto, nei dettagli. Nella Home l'utente può ricercare articoli in base a parole chiave contenute nell'*abstract*, in base a parole chiave contenute nel titolo, in base all'autore, e nella versione presentata in questa

[11] Online in development mode su http://eelst.cs.unibo.it:8088/#/app/homeSearch, in produzione su http://eelst.cs.unibo.it:8089/#/app/homeSearch, su GitHub https://github.com/alicegraziosi/BEX

[12] Attività di ottenere informazioni a partire da un contesto

tesi in base alla *venue* di appartenenza. Di default i risultati in output vengono presentati secondo l'anno di pubblicazione in ordine decrescente. L'utente può scegliere di ordinare i risultati anche per numero di citazioni, in ordine crescente oppure decrescente. Per ogni articolo vengono riportate le informazioni essenziali che lo caratterizzano: titolo, lista di autori, tipologia, anno di pubblicazione, nome e dettagli della rivista di appartenenza. In aggiunta è presente il link alla risorsa dell'articolo, il link esterno alla pagina dell'articolo sul sito web ufficiale dell'editore e bottone per il *bookmark*. L'abstract, il DOI e i dettagli sulle citazioni, divise per *incoming* e *outgoing*, sono visibili a scelta dell'utente in una tendina a scomparsa. Nel box sezione *incoming* sono indicati il numero di citazioni locali contate all'interno del dataset, e il numero di citazioni globali calcolate da servizi esterni. Nel box sono visualizzati tutti gli articoli citati/citanti e i loro dati di base, insieme con un grafico a torta che riassume il numero di volte in cui un articolo è citato/cita diviso per i motivi citazionali, quando previsti dalla struttura del dataset. Per ogni articolo c'è un menù popup che mostra le frasi dell'articolo corrente in cui è stato citato. L'applicazione permette di navigare gli articoli attraverso le citazioni in virtù del fatto che per ogni articolo citato/citante, si può andare alla pagina di dettaglio di quell'articolo. Una particolatà del sistema è quella di indicare graficamente quando un articolo è autoreferenziale cioè quando cita o è citato da articoli scritti da uno o più autori in comune con l'articolo corrente.

Per valutare BEX in termini di efficienza e usabilità è stato condotto un test con gli utenti. L'analisi è stata fatta direttamente sul sito da parte degli utilizzatori a cui è destinata l'applicazione, cioè principalmente docenti e ricercatori universitari. I task da compiere richiesti sono stati: trovare articoli recenti su un dato argomento, controllare quanto un articolo fosse aggiornato, quanto autoreferenziale, e quanto d'impatto. I rapporti e le opinioni che provengono dal questionario sulla soddisfazione dell'utente hanno portato un riscontro positivo, ma hanno fatto emergere alcune mancanze. Alcune di esse sono state colmate dalla nuova versione, altre sono ancora considerate come sviluppi futuri.

Capitolo 3

BEX: estensioni e integrazioni

Si ritiene utile e interessante continuare a implementare nuove funzionalità e garantire la manutenzione di BEX in quanto strumento per il sense-making di informazioni contenute in datasets altrimenti difficilmente navigabili. Inoltre, essendo collocato all'interno di un'area di ricerca emergente e in costante evoluzione come lo è il Semantic Publishing, è particolarmente sentita l'esigenza di apportare continui miglioramenti. Partendo dagli sviluppi futuri individuati per la versione precedente di BEX, si è deciso quali contributi dovessero essere introdotti per ovviare ai problemi del sistema. La prima versione dell'applicazione prevedeva la possibilità di scegliere quale dataset interrogare, ma non era stato implementato il modo di far girare l'app con basi di dati diverse. Era infatti pensata *ah hoc* solo per SLT, un solo tipo di dataset dotato di una certa struttura, inoltre non c'era separazione tra codice e query. Il numero massimo di articoli recuperati tramite query era limitato a un numero di cinquanta dato che non c'era modo di visualizzare un gran numero di risultati in pagina senza dover scorrere troppo la schermata e dato che l'esecuzione delle query e l'elaborazione dei dati sono dispendiose in termini di tempo. Porre rimedio a variazioni in termini sia di volume che di struttura del contenuto dei dataset RDF è decisivo per la nuova versione di BEX.

Per decidere quali dovessero essere le integrazioni da apportare al sistema si è partiti dagli sviluppi futuri della prima versione dell'applicazione e dalle carenze delle DL esistenti. Le maggiori integrazioni introdotte in BEX sono:

– l'indipendenza dal dataset SLT al fine di predisporre l'app all' uso di altri datasets;

– la paginazione dei risultati di ricerca e la scelta del numero massimo di risultati visibili per pagina;

– l'aggiunta dei dati aggregati sul singolo autore, affinchè si abbiano viste *author-centric*;

– l'integrazione di Citation Explorer, per la comparazione di due autori;

– la ricerca in base alla venue di appartenenza degli articoli e la ricerca nome completo dell'autore.

3.1 Indipendenza dal Semantic Lancet Triplestore

Come detto in precedenza, originariamente BEX era basato solo sul dataset SLT. Data l'importanza e la diffusione a scopi di ricerca degli scholarly Linked Open Dataset, si è rivelato necessario rendere l'app BEX indipendente dal dataset di Semantic Lancet e utilizzabile con altri dataset potenzialmente strutturati diversamente da SLT. Non tutti i dataset modellano infatti le stesse informazioni con le stesse ontologie ed è perciò fondamentale fare lo sforzo di adattare l'applicazione a poter elaborare quanti più datasets diversi possibili. Nella sezione Settings dell'applicazione è possibile scegliere il dataset da analizzare e il sistema indirizza le queries all'endpoint dataset appropriato per ciò che è stato selezionato. A oggi l'app è stata testata con un dataset contenente metadati RDF di una parte delle pubblicazioni di Springer[1]. I metadati del dataset preso in esame includono informazioni su quasi 200 venue, 3000 autori e 800 articoli, numeri molto più alti rispetto a quelli di SLT, e che sono ancora maggiori in altri dataset disponibili come Open Linked Data che potranno essere esaminati con BEX, tra cui il repository Open Citations Corpus (OCC) di cui si parlerà pià approfonditamente nella sezione sviluppi futuri in conclusione. Il dataset di Springer è molto diverso dal dataset SLT, non contiene infatti informazioni sull'abstract degli articoli e nè i contesti citazionali nè i motivi citazionali, ma contiene metadati di articoli di diverso tipo rispetto a SLT. Gli articoli di Springer possono far parte non solo di Journal, ma anche di Book, e sono JournalArticle, BookChapter, ProceedingsPaper o AcademicProceedings. Il vasto numero di dati

[1] http://www.springer.com/us/

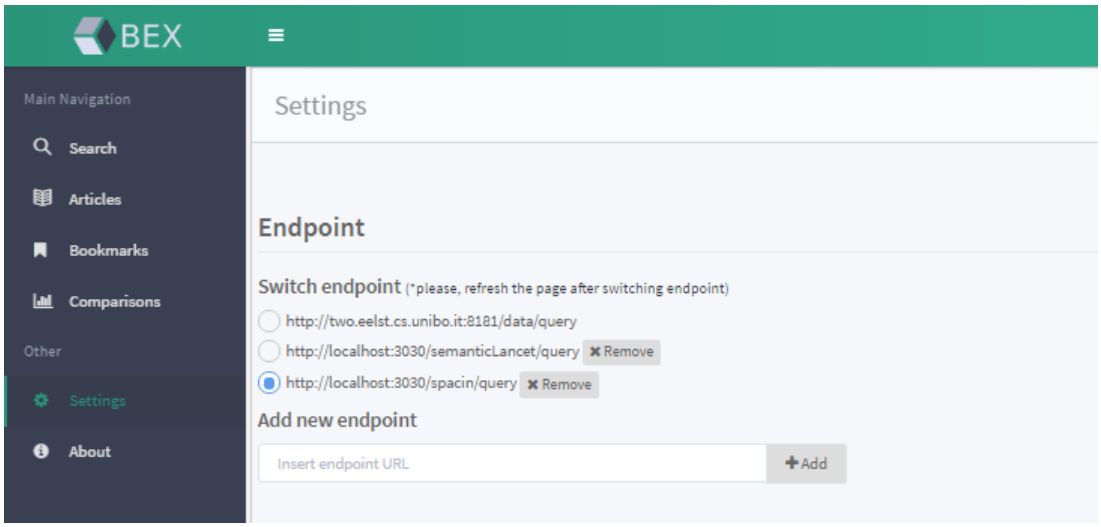

Figura 3.1: Scelta del dataset da navigare

contenuti nei dataset è il motivo per cui è stato necessario introdurre un meccanismo di paginazione descritto nel paragrafo seguente. Una delle integrazioni maggiori apportate a BEX è infatti la paginazione di risultati, indispensabile quando le ricerche vengono fatte su un grande quantitativo di dati e capita frequentemente di ottenere un numero molto alto di risultati.

3.2 Paginazione dei risultati

3.2.1 Paginazione

Per implementare la paginazione dei risultati è stata usata la stessa logica del motore di ricerca Google[2]. La logica di paginazione prevede che vengano visualizzate dieci pagine alla volta, a meno che le pagine totali siano meno di dieci. Il link attivo è sempre quello alla posizione sei, fatta eccezione per il caso in cui la pagina corrente è quella a posizioni minori di sei o minori di quattro a partire dall'ultima posizione. Grazie alla paginazione si risolve il problema che si manifesta in presenza di dataset di grandi dimensioni, cioè

[2]https://www.google.it/

quando le ricerche fatte dall'utente possono portare a un alto numero di risultati in pagina.

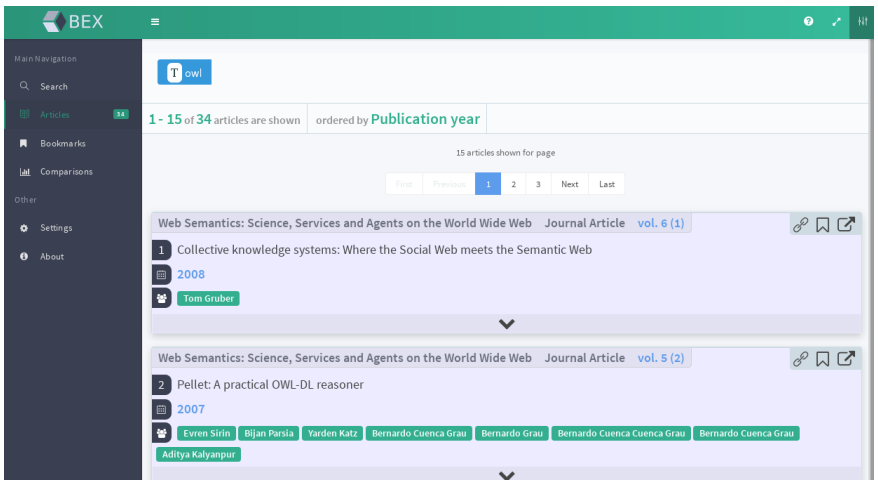

Figura 3.2: Esempio paginazione: pagina 1 di 3

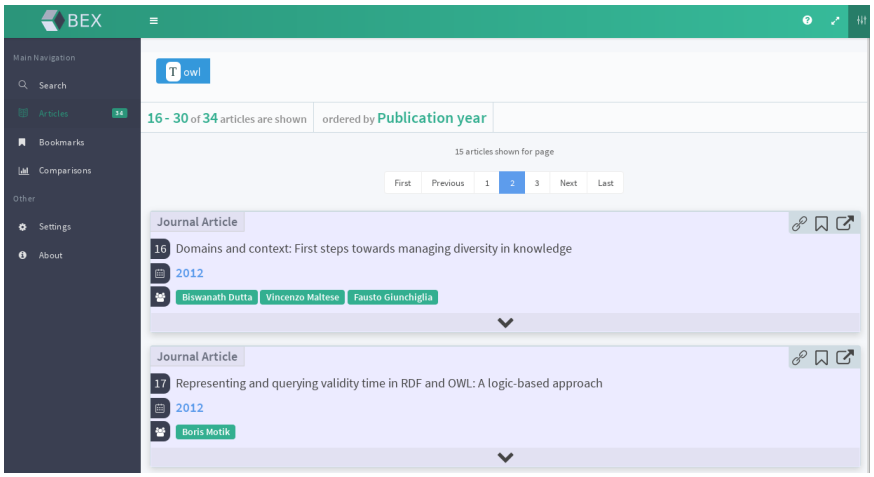

Figura 3.3: Esempio paginazione: pagina 2 di 3

3.2.2 Numero massimo di risultati per pagina

Il numero massimo di risultati visibili per pagina è variabile e a scelta dell'utente. L'utente può decidere nella sezione Settings il numero massimo di risultati che desidera

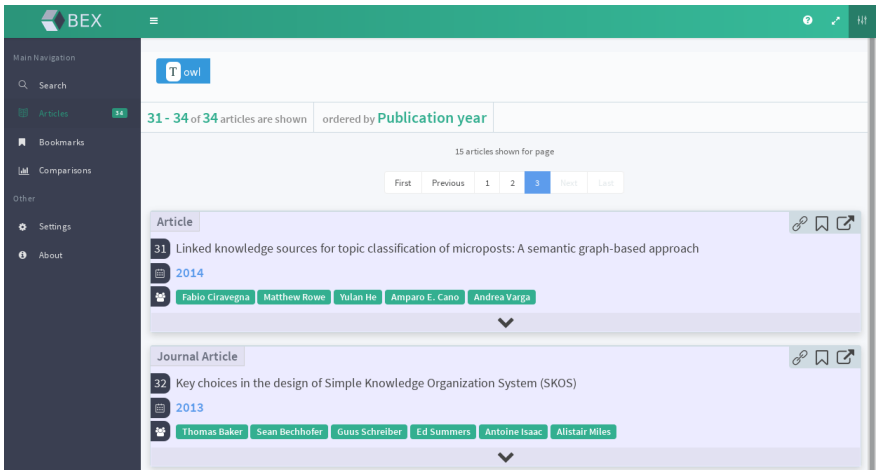

Figura 3.4: Esempio paginazione: pagina 3 di 3

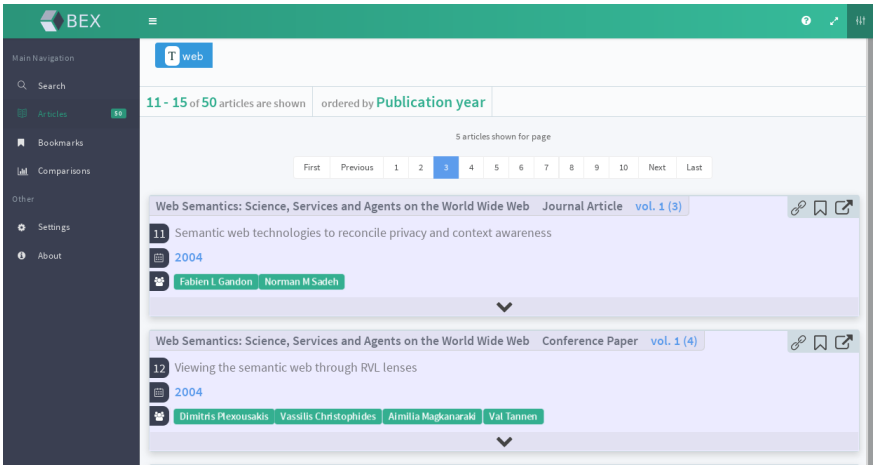

Figura 3.5: Esempio paginazione: 10 pagine

visualizzare in pagina. Il numero massimo di default se non diversamente impostato è di 30 risultati. Il numero di pagine totali è dato dalla quantità totale di risultati divisa per il numero massimo di risultati visibili per pagina.

3.3 Dati aggregati sul singolo autore

La versione precedente dell'applicazione non prevedeva viste incentrate sui dati di un singolo autore. Si è così voluto aggiungere alcune viste *author-centric* aggregando i

Figura 3.6: Scelta del numero massimo di risultati per pagina

dati del singolo autore. Quando l'utente ricerca gli articoli di un particolare autore nella schermata dei risultati vengono visualizzate due sezioni. La prima sezione offre una sintesi delle informazioni che si riescono a reperire sull'autore, la seconda mostra semplicemente i papers dell'autore. L'overview ottenuta con l'aggregazione dei dati contiene il numero totale di citazioni ricevute trovate all'interno del singolo dataset, il numero di citazioni globali in entrata, le statistiche e la comparazione con un altro autore. Le statistiche illustrano graficamente quante volte e perché un autore è stato citato, sia in generale che secondo una divisione per anno. Infine si può cercare un altro autore con cui fare un confronto e visualizzare come cambiano nel tempo il numero e i motivi citazionali, quando presenti, nella struttura del dataset interrogato, di citazioni in entrata e in uscita.

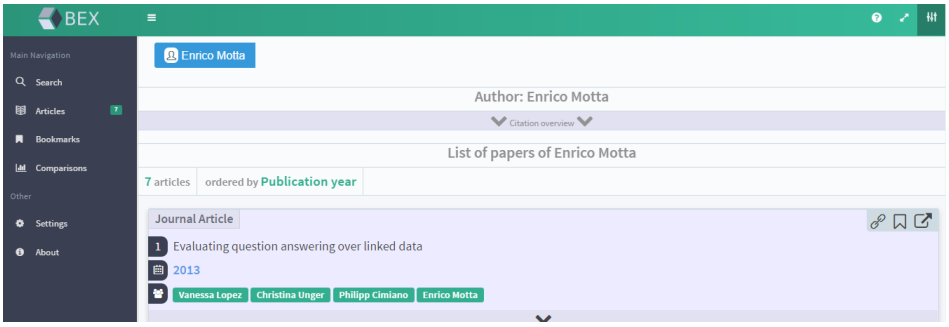

Figura 3.7: Pagina dei risultati di articoli di un autore

3.4 Integrazione di Citation Explorer

Un'altra interessante vista *author-centric* è il confronto tra autori in base alla distribuzione del numero di citazioni in entrata e in uscita divise per anno. È stata dunque

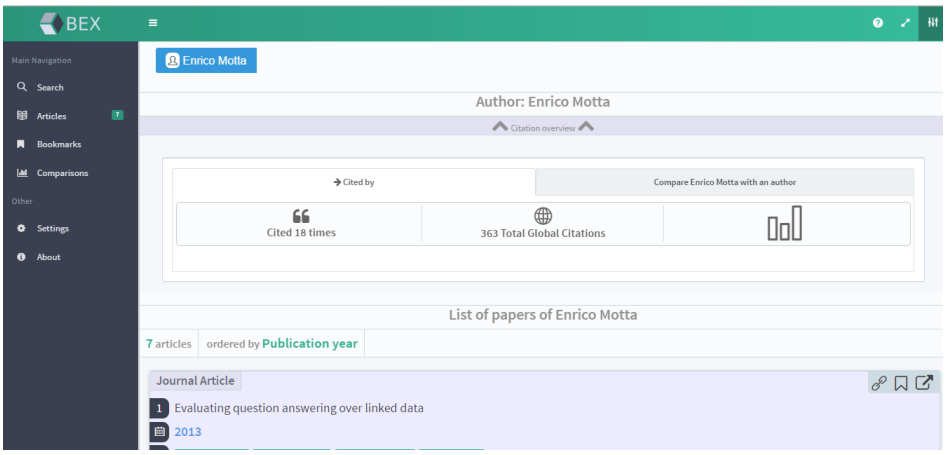

Figura 3.8: Overview dei dati aggregati di un autore

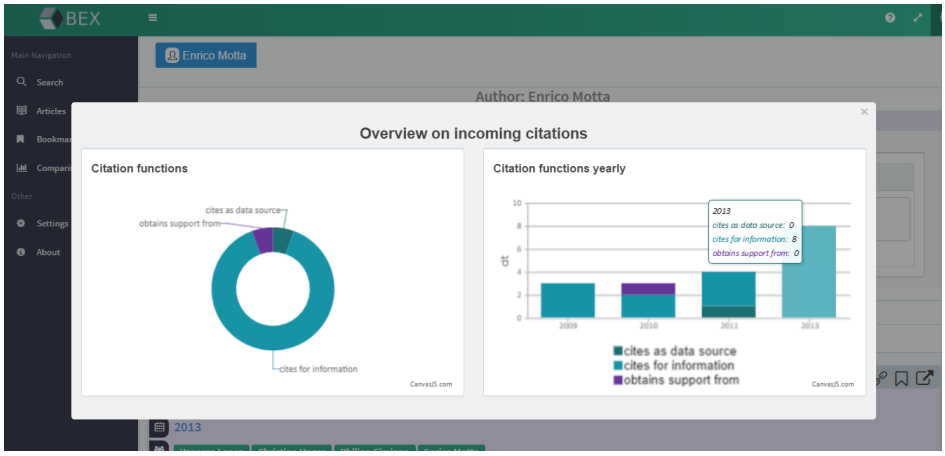

Figura 3.9: Citazioni ricevute trovate nel dataset corrente, citazioni globali, statistiche

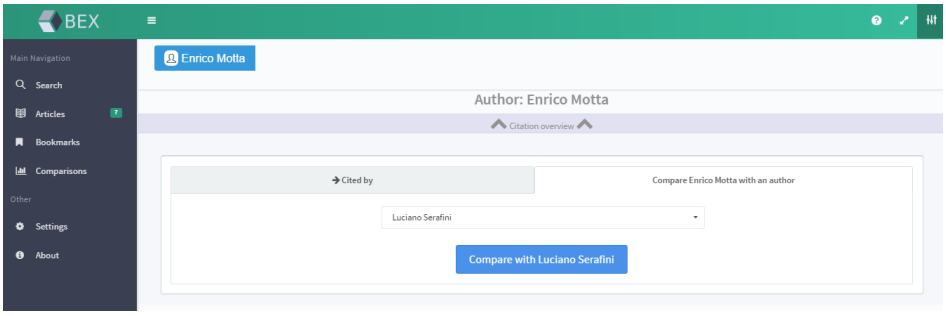

Figura 3.10: Schermata di comparazione con un altro autore

integrata a BEX una parte del tool web-based Citation Explorer[3] facente parte del progetto Semantic Lancet. La sezione Comparison è stata aggiunta al menù laterale. Questa integrazione ha come scopo il sense-making delle citazioni, tramite un grafico a barre che permette all'utente di contrapporre l'attività di due autori mostrando come variano il numero e la tipologia di citazioni in entrata e in uscita col passare del tempo.

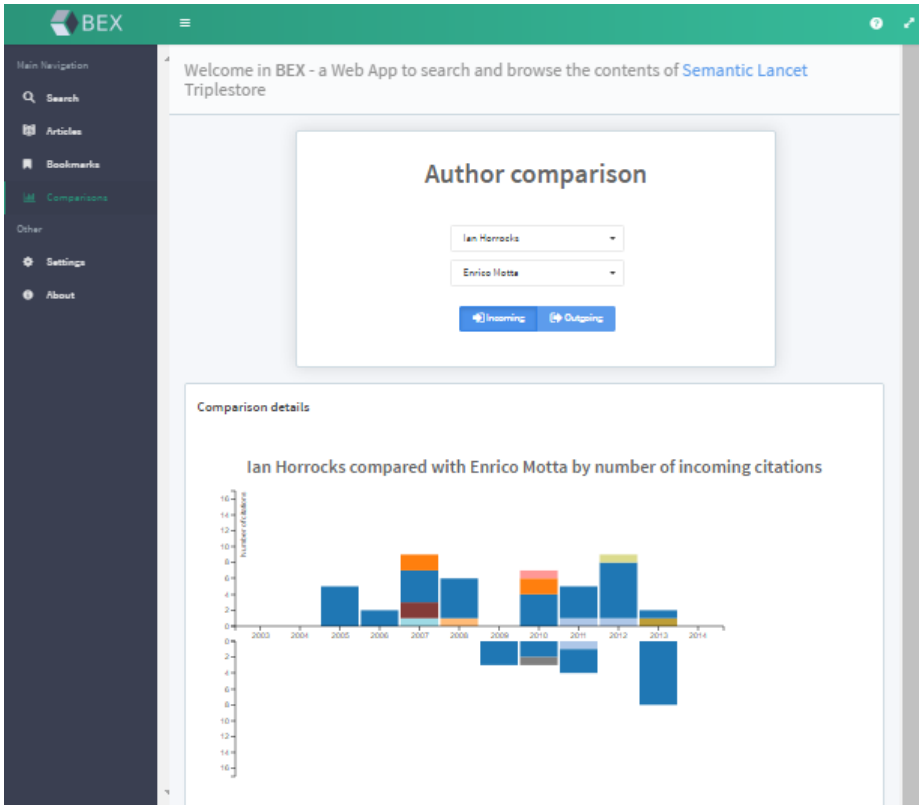

Figura 3.11: Grafico della distribuzione delle citazioni in entrata di due autori

[3]http://www.semanticlancet.eu/citationexplorer/index.php

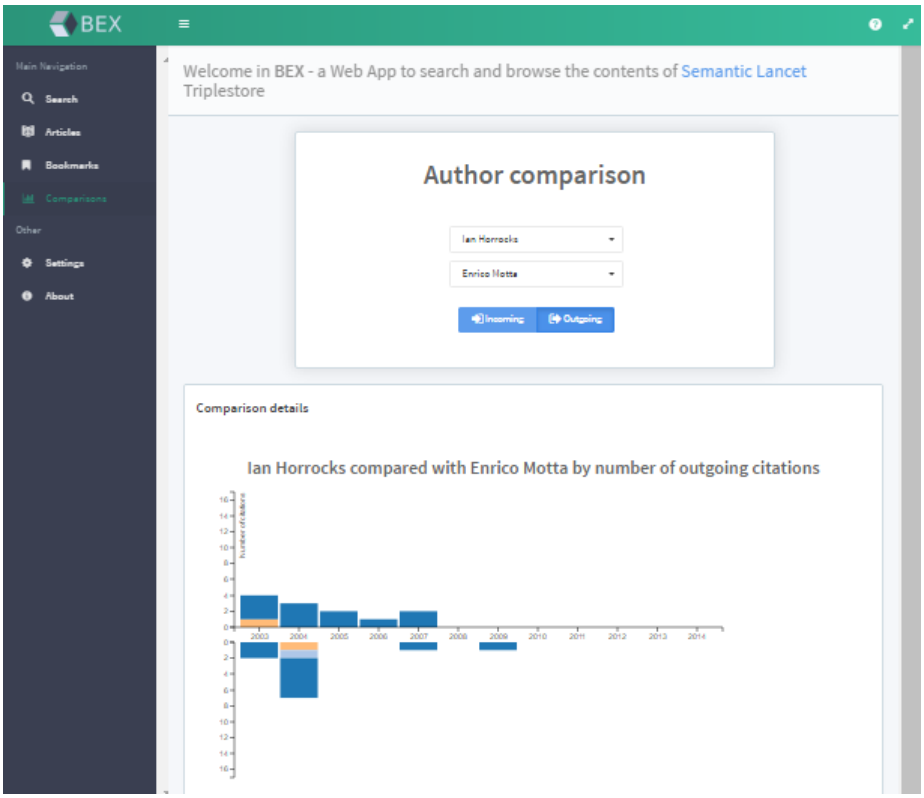

Figura 3.12: Grafico della distribuzione delle citazioni in uscita di due autori

3.5 Ricerca in base alla venue di appartenenza degli articoli e ricerca per nome completo dell'autore

3.5.1 Ricerca in base alla venue di appartenenza degli articoli

Nel Semantic Lancet Triplestore sono presenti solo i metadati delle pubblicazioni della sola rivista Journal of Web Semantics di Elsevier, e dunque non era stata presa in considerazione l'implementazione della ricerca in base alla venue di appartenenza degli articoli. I metadati presenti nel dataset di Springer riguardano invece articoli pubblicati in più di una rivista tecnica. Questo motivo ha portato allo sviluppo della ricerca per venue, oltre al fatto che è utile per un affinamento della ricerca in generale.

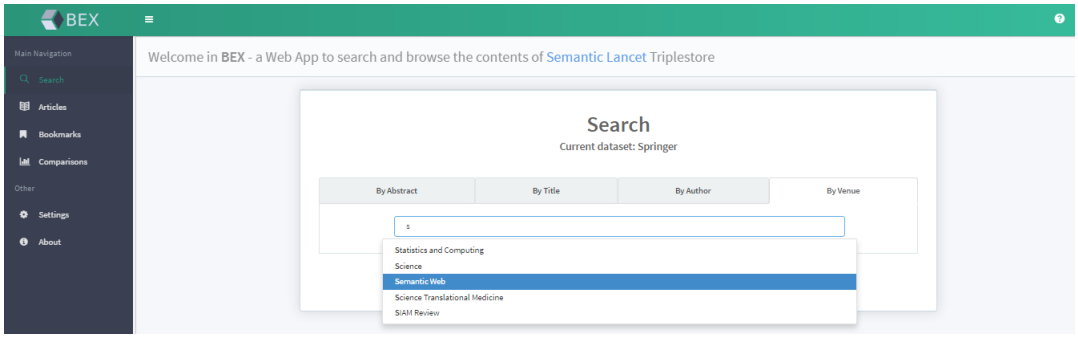

Figura 3.13: Modalità di ricerca in base alla venue di un articolo

3.5.2 Ricerca per nome completo dell'autore

Una piccola modifica sulle modalità di ricerca riguarda la ricerca in base al nome completo dell'autore. Nella versione precedente di BEX si poteva solo scorrere tutta la lista degli autori o sceglierne uno ricercandolo per nome. Per una questione di comodità e di memoria delle persone invece è più semplice cercare un autore in base a ciò che si ricorda del suo nome intero.

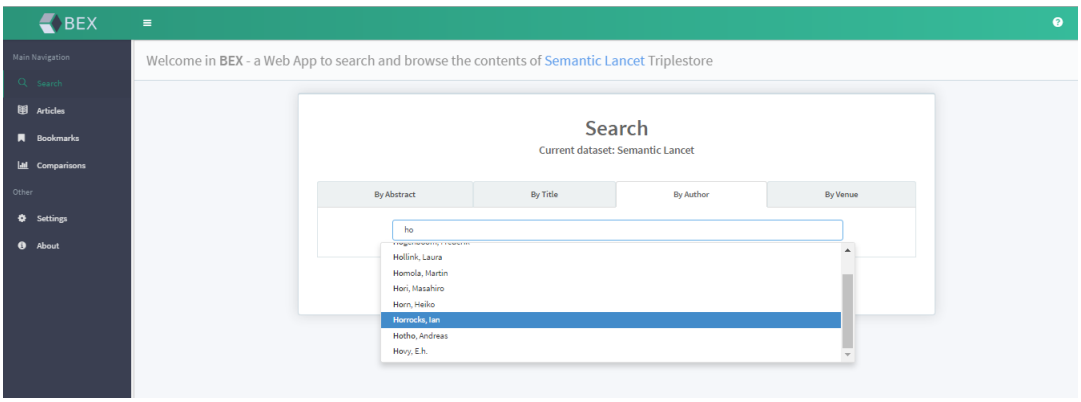

Figura 3.14: Modalità di ricerca in base al nome completo dell'autore

Capitolo 4

Dettagli implementativi

4.1 Tecnologie utilizzate

L'intera applicazione è sviluppata con AngularJS[1], un framework JavaScript open source fornito da Google con l'obiettivo di semplificare l'implementazione di single-page application basato sull'architettura MVC (Model View Controller). Le altre tecnologie web utilizzate congiuntamente ad AngularJS sono HTML, CSS, Bootstrap[2] e altre librerie JavaScript come AngularUI Router[3] e BootstrapUI[4]. L'uso di AngularJS e la progettazione di BEX come single-page application rende l'app interattiva e consente una buona usabilità. È opportuno notare che in questa nuova versione di BEX, come nella precedente, non c'è stato bisogno di particolari script lato server. Non c'è nessun componente server-side, tranne che per il pacchetto npm[5] http-server[6], un server HTTP configurabile a riga di comando per NodeJS[7] usato esclusivamente per caricare l'intera applicazione. Tutte le richieste vengono fatte lato client a server esterni o a SPARQL endpoint tramite le REST API che forniscono.

[1] https://angularjs.org/, https://angular.io/
[2] http://getbootstrap.com/2.3.2/
[3] https://angular-ui.github.io/ui-router/site/#/api/ui.router
[4] https://angular-ui.github.io/bootstrap/
[5] https://www.npmjs.com/
[6] https://www.npmjs.com/package/http-server
[7] Node.js, https://nodejs.org/en/, è un framework javascript lato server.

4.1.1 AngularJS

AngularJS è un potente framework web client-side particolarmente utile per creare single-page application con viste dinamiche realizzabili estendendo il linguaggio HTML. Angular implementa il pattern MVC e dunque permette il disaccoppiamento della manipolazione dal DOM dalla logica applicativa. Offre funzionalità a supporto dello sviluppatore per creare web application dall'architettura modulare e di produrre codice leggibile in maniera veloce. Data la sua natura modulare è facilmente estendibile e lavora bene in composizione con molte librerie esterne. I moduli di Angular sono principalmente *controllers*, *services*, *filters* e *directives*.

Two Way Data-Binding

Il concetto di *Two Way Data-Binding* descrive una situazione in cui: quando cambia lo stato l'oggetto che rappresenta il *model* allora si aggiorna anche l'interfaccia grafica; e quando vengono fatte modifiche sull'intefaccia utente, queste si trasmettono anche al *model*. Il data-binding viene implementato tramite componenti di tipo *Expression*. AngularJS è dunque un framework *data-driven* in cui la vista e il modello sono sincronizzati: per mantenere aggiornate le informazioni da visualizzare nell'interfaccia basta instaurare un binding tra il model e la view e al variare del modello verrà automaticamente modificata l'interfaccia. Con AngularJS le pagine HTML diventano templates dinamici.

Pattern MVC

MVC (Model View Controller)[8] è un pattern architetturale tramite cui si è in grado di separare la *presentation logic* (interfaccia utente) dalla *business logic* (logica applicativa), garantendo una solida distinzione delle responsabilità in oggetti indipendenti che rendono l'applicazione modularmente suddivisa in livelli. Il pattern suddivide i componenti software in base ai ruoli che hanno all'interno di un sistema:

- **model**: sono i dati che gestisce il sistema e che l'utente vede nell'interfaccia grafica (in BEX sono oggetti JSON);

[8] https://it.wikipedia.org/wiki/Model-View-Controller

- **view**: visualizza i dati formalizzati nel *model* e si occupa della *user interaction*, ha dunque in carico il ruolo di renderizzare l'interfaccia utente (in BEX sono pagine HTML customizzate);

- **controller**: riceve i comandi dell'utente attraverso la *view* e modifica lo stato degli altri due componenti, gestisce i dati del modello e la loro rappresentazione (in BEX sono componenti JavaScript di AngularJS).

Il modulo *view* ha in carico l'interfaccia utente, mentre il modulo *model* e quello *controller* si occupano della logica applicativa.

Dependency injection

Dependency injection (DI) è un design pattern dell'ingegneria del software che gestisce il modo in cui i componenti da cui un certo modulo dipende vengono forniti a quel modulo. L'oggetto *injector* fornisce a tempo di esecuzione le dipendenze di un controller o servizio.

Expression

Un'*expression* Angular è simile a un'espressione JavaScript. Vengono usate all'interno del tag HTML per implementare il data-binding e quindi interpolare gli elementi delle pagine Web con i dati da visualizzare. Inoltre possono essere utilizzate, congiuntamente con le direttive, per allacciare direttamente a un elemento il codice che deve essere eseguito all'accadere di un evento.

Scope

Uno *scope* è un oggetto che si riferisce al *model* e fornisce il contesto di esecuzione delle *expressions*. Gli scopes sono ordinati in un'organizzazione gerarchica che simula la struttura DOM[9] dell'applicazione. Si possono annidare gli scopes per limitarne la visibilità e l'accesso da parte dei componenti dell'applicazione Angular. Per tenere sotto

[9]Document Object Model

controllo le variazioni del model a cui è associato uno scope, e per propagare i cambiamenti del model attravverso le altre componenti Angular e per aggiornare una view si può usare il *listener* `$watch`.

Controller

Per ogni *view* c'è un *controller* che ne gestisce la logica applicativa. Un controller è aggiunto al modulo generale dell'applicazione (`angular.module()`) tramite il costruttore `.controller()` e per ognuno di essi viene creato uno scope figlio. Un controller è attaccato al DOM attraverso la direttiva `ng-controller` e dovrebbe riferirsi al comportamento di una sola view.

Directives

Le *directives* sono markers da inserire sulla dichiarazione di un elemento del DOM e che vengono individuate dal compilatore HTML che associa il comportamento specificato all'elemento rendendo la pagina HTML interattiva. AngularJS si presenta con molte directives *built-in* (come `ngBind`, `ngModel`, `ngClass`), ma è possibile crearne di personalizzate. Il nome del file .js che contiene il codice della directive è normalizzato in *camelCase* mentre per richiamare la direttiva dentro a un tag HTML si usa la forma delimitata da trattino. Per registrare una directive si usa la module.directive API seguita da una *factory function*. La factory function viene invocata quando il compilatore HTML la appaia con l'elemento a cui è associata.

Service

Gli Angular *services* sono oggetti collegabili tra loro tramite Dependency Injection (DI), riusabili in più parti dell'applicazione quante volte si vuole. Per utilizzare un service Angular bisogna aggiungerlo tra le dipendenze di un componente (controller, service, filter o directive) che dipende dal service e il sottosistema di Angular per la Dependency Injection si occuperà del resto. Angular istanzia un service una sola volta appena un componente dell'app ne ha dipendenza, e ogni componente che ne dipende ottiene un riferimento alla singola istanza del servizio creata dalla service factory.

4.1.2 D3.js

D3.js Data-Driven Documents[10] è una libreria JavaScript che consente di manipolare documenti basati su dati. D3.js sfrutta gli standard del web HTML, SVG[11] e CSS per la generazione dinamica di visualizzazioni elaborate. La librerie offre un approccio *data-driven* per manipolare il DOM e permette di costruire oggetti grafici complessi. Per gli scopi dell'applicazione è stata usata per creare grafici a barre a partire da dati arbitrari.

4.2 Dettagli

4.2.1 Indipendenza dal Semantic Lancet Triplestore

Per implementare l'indipendenza del sistema dal dataset SLT è prima di tutto necessario separare la lista di queries dalla struttura del sistema e inserirle in diversi file separati. A ogni dataset, o a più dataset similmente strutturati, corrisponde un file di queries appositamente costruite. Per poter adattare l'applicazione a interagire con datasets diversi bisogna riscrivere un numero di circa 20 query, quando non si è in presenza di contesti e funzioni citationali, e 25 quando sono invece previsti dalla struttura del dataset. Gli id degli script che contengono una singola query devono essere mantenuti uguali, così come anche deve essere mantenuta la lista di output che deve essere restituita. L'insieme di queries deve essere contenuta interamente in un file da censire infine nel file di configurazione `endpointQueryFile.json`. Nella sezione Settings di BEX l'utente è abilitato a scegliere di quale dataset vuole visualizzare i dati e, in base alla scelta fatta, il sistema carica il file di query appropriato. La corrispondenza tra il nome del dataset, l'indirizzo dello SPARQL endpoint e il nome del file di queries da caricare si trova in un file JSON.

```
[
  {
    "endpoint": "http://two.eelst.cs.unibo.it:8181/data/query",
    "queryFile": "queriesSemanticLancet.html",
```

[10] https://d3js.org/, https://github.com/d3/d3/wiki/Gallery
[11] Scalable Vector Graphics, https://www.w3.org/Graphics/SVG/

```
    "datasetName": "Semantic Lancet"
  },
  {
    "endpoint": "http://localhost:3030/semanticLancet/query",
    "queryFile": "queriesSemanticLancet.html",
    "datasetName": "Semantic Lancet"
  },
  {
    "endpoint": "http://localhost:3030/spacin/query",
    "queryFile": "queriesSpringer.html",
    "datasetName": "Springer"
  }
]
```

Dopo che l'utente ha scelto quale dataset vuole interrogare è necessario ricaricare l'intera pagina in modo che venga caricato insieme anche il file delle query adatto. Appena il *loading* di questo file è completato viene inizializzato tutto l'ambiente con i dati provenienti dal dataset.

4.2.2 Il dataset di Springer

La struttura del dataset delle pubblicazioni della rivista Springer è diversa da quella secondo cui sono modellati i dati del Semantic Lancet Triplestore, pur essendo utilizzate le stesse ontologie, le informazioni sono presentate in forme diverse e non tutti i dati che sono descritti in uno lo sono anche nell'altro. Le differenze tra le due collezioni di record iniziano dall'uso dell'ontologia FaBiO (FRBR-aligned Bibliographic Ontology).

Le principali entità del dominio delle pubblicazionni scientifiche modellate dall'ontologia FaBiO sono:

- **Work**: una creazione intellettuale;
- **Expression**: una realizzazione di un *Work*;
- **Manifestation**: la materializzazione di un'*Expression*;

– **Item**: un esemplare di una *Manifestation*.

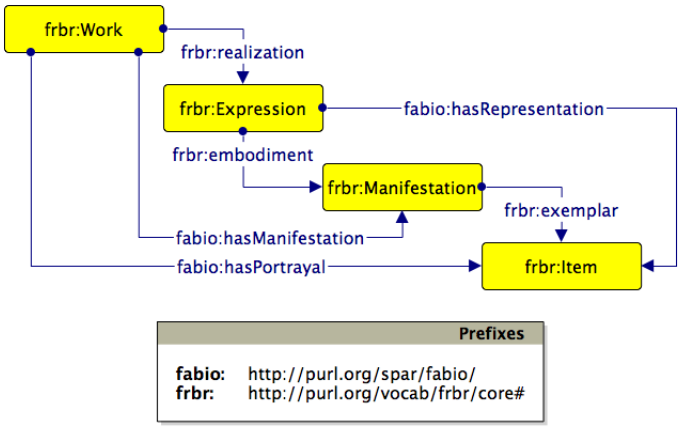

Figura 4.1: Diagramma Graffoo[27] del modello dei dati FRBR esteso da FaBiO con proprietà addizionali. Fonte: http://www.sparontologies.net/ontologies/fabio

In SLT la modellazione dei dati parte dall'entità *Work*, mentre nel dataset di Springer si comincia dall'entità *Expression*. A differenza di SLT, gli autori di un articolo non sono strutturati come una lista di *Person*, ma come singoli *Agent* che hanno il ruolo di autore per un dato documento. Nel dataset di Springer non ci sono statements RDF nè sull'*abstract* di un articolo, nè sui dettagli riguardanti i riferimenti bibliografici. Interrogando il dataset si possono solo conoscere quali articoli sono citati in un dato articolo e viceversa da quali articoli è citato un articolo, ma non è specificato nessun dettaglio sulle relazioni tra essi. Invece in SLT è indicata anche la posizione in cui si trovano le citazioni all'interno del testo di un paper (*citation context*) e sono dichiarati i motivi per cui un articolo cita un altro (*citation function*). Per gestire le differenze nell'uso delle ontologie e rendere l'applicazione svincolata dalla conformazione di un dataset è stato necessario adeguare il codice preesistente del *service* Angular `articles-manager.js`, modulo che si occupa del recupero delle informazioni derivate dalle queries preparandole per il *controller* `articles-results.js` apposito per la visualizzazione dei risultati in pagina. In più si è dovuto scrivere un file di queries corenti con il nuovo dataset da caricare insieme all'intera applicazione. Per datasets cotruiti secondo strutture ancora differenti sarà necessario riformulare di nuovo le queries e aggiungere il file all'applicazio-

ne. L'interfaccia che mostra i risultati di una ricerca è la stessa per entrambe le versioni, tranne per il fatto che alcuni bottoni devono essere disabilitati e alcuni grafici non sono presenti (grafici di riepilogo sui dettagli delle citazioni).

Di seguito due delle queries più significative.

Listing 4.1: Lista delle pubblicazioni di un dato autore per il dataset di Springer

```
SELECT ?expression
WHERE {
    ?agent a foaf:Agent;
        foaf:givenName 'givenName';
        foaf:familyName 'familyName';
        pro:holdsRoleInTime ?role.

    ?role a pro:RoleInTime;
        pro:withRole pro:author;
        pro:relatesToDocument ?expression.
}
```

Listing 4.2: Dettagli di un articolo (di tipo JournalArticle o BookChapter) per il dataset di Springer

```
SELECT DISTINCT ?journalTitle ?title ?doi ?expression
                                                    ?volumeN
                                                    ?htmlIte
WHERE {
    <expression> a fabio:Expression;
        dcterms:title ?title;
        fabio:hasPublicationYear ?publicationYear.
    OPTIONAL{
        <expression> frbr:partOf [ a fabio:Book;
            dcterms:title ?journalTitle;
            datacite:hasIdentifier [
```

```
        a datacite:Identifier;
            datacite:usesIdentifierScheme datacite:isbn;
            litre:hasLiteralValue ?bookIdVal;
        ]
    ].
}
OPTIONAL{
    <expression> frbr:partOf
                [ a fabio:JournalIssue, fabio:Expression;
            datacite:hasIdentifier ?issueIdentifier;
            fabio:hasSequenceIdentifier ?issueNumber;
            frbr:partOf
                [ a fabio:JournalVolume, fabio:Expression;
                datacite:hasIdentifier ?volumeIdentifier;
                fabio:hasSequenceIdentifier ?volumeNumber;
                frbr:partOf [ a fabio:Journal;
                    dcterms:title ?journalTitle;
            ]
        ]
    ].
}
OPTIONAL{
    <expression> frbr:embodiment [ a fabio:Manifestation;
        prism:startingPage ?startingPage;
        prism:endingPage ?endingPage;].
}
OPTIONAL{
    <expression> datacite:hasIdentifier
            [ a datacite:Identifier;
        datacite:usesIdentifierScheme datacite:occ;
        litre:hasLiteralValue ?bookChapterIdVal;].
```

```
    }
    OPTIONAL{
        <expression> datacite:hasIdentifier
                [ a datacite:Identifier;
            datacite:usesIdentifierScheme datacite:doi;
            litre:hasLiteralValue ?doi;].
    }
    OPTIONAL{
        <expression> datacite:hasIdentifier [
            a datacite:Identifier;
            datacite:usesIdentifierScheme datacite:url;
            litre:hasLiteralValue ?htmlItem].
    }
}
```

4.2.3 Paginazione dei risultati

Pager Service

La logica di paginazione è stata implementata in `pager-service.js`, un service di AngularJS. Non appena il *controller* che si occupa della visualizzazione in pagina dei risultati della ricerca inizia a ricevere gli articoli risultati viene invocato il servizio di paginazione. In presenza di un numero alto di risultati la versione precedente del sistema impiegava tempi troppo alti e spazi troppo ampi dell'interfaccia per la loro visualizzazione in pagina ed era stato dato un limite massimo di risultati che una query potesse restituire. Con la paginazione invece non ci sono problemi in termini di spazio utile occupato nell'interfaccia utente nel caso in cui siano molti gli articoli da mostrare. La soluzione è ancora migliorata dalla possibilità di scegliere il numero massimo di risultati visibili in una sola pagina. Anche con la paginazione però, il tempo che serve per caricare tutti i dati in pagina è ancora alto e l'utente non si trova ancora davanti a un'interfaccia caricata in tempi ragionevoli. Si è così deciso di spezzare la gestione dei dati quando si supera il numero di 30 risultati totali. Se i risultati della ricerca sono maggiori di 30

allora vengono divisi in due sezioni, i primi vengono imemdiatamente inviati al controller competente che inizia a paginare. La barra di loading in alto si completa quando arrivano i primi 30 risultati e poi scompare, dando l'idea all'utente che non ci siano dati in caricamento. In realtà i dati stanno ancora arrivando in background, ma quelli disponibili sono già visualizzati in modo che i tempi di visualizzazione siano ragionevoli anche in presenza di tempi di calcolo lunghi.

Numero massimo di risultati per pagina

Nella sezione Settings l'utente può scegliere il numero di risultati che vuole caricare in una singola pagina. Senza bisogno di ricaricare la pagina, quando si effettua una nuova ricerca verranno costruite tante pagine quanti sono il numero di risultati divise per il numero scelto.

4.2.4 Dati aggregati sul singolo autore

Un aspetto che mancava nella vecchia versione dell'applicativo e che invece è spesso fornita dalle Digital Libraries più complete allo stato dell'arte era una visualizzazione *author-centric* dei dati. Viste di questo tipo sono utili a giudicare la produttività di un autore e la sua popolarità all'interno della comunità scientifica. Quando si fa una ricerca per autore nella nuova versione di BEX, oltre alla lista dei papers di un autore, viene anche mostrato un' anteprima delle informazioni che si sono potute ottenere dall'aggregazione dei dati. L'aggregazione dei dati si fa prendendo in considerazione tutti gli articoli di un autore e calcolando il totale delle citazioni ricevute e quelle in uscita, sia quelle ottenute computando i dati nel dataset sia quelle globali. Non c'è stato bisogno di aggiungere nuove queries alla lista, le informazioni sulle citazioni degli articoli vengono calcolate a partire dai dati recuperati dai paper risultati dalla ricerca sul dataset. Ciò che è stato necessario è l'aggiunta di un template HTML per la visualizzazione dei dati `author-result.html`.

4.2.5 Integrazione di Citation Explorer

Come anticipato Citation Explorer è un tool per il *sense-making* dei riferimenti bibliografici in un'ottica *author-centric*. Nel menù laterale è stata aggiunta una nuova voce di menù *Comparison* e nell'overview dei risultati restituiti dalla ricerca per autore è presente una sezione che permette di scegliere un autore con cui comparare quello corrente. Per integrare il tool è stato adattato il codice secondo la struttuta generale del sistema in modo da mantenere l'applicazione componibile. Le queries sono state riscritte e aggiunte ai file appropriati e si è cercato separare e riusare le variabili e i moduli preesistenti. La comparazione della distribuzione delle citazioni è stata implementata tramite un template dinamico *dh-visualization.html* e il corrispondente componente Angular di tipo *directive* dal nome `dh-visualization.js`. Questo elemento è invocato, senza averne dovuto fare modifiche, anche nella vista di comparazione che appare nella schermata dei risultati della ricerca per autore. Utilizzare framework come AngularJS infatti porta a risparmiare tempo nella scrittura di un programma in quanto è progettato per favorire il riuso di codice all'interno di una stessa applicazione. Tramite grafici costruiti con la libreria D3.js viene mostrato il variare del numero delle citazioni in entrata e in uscita per i due autori in esame. L'andamento del numero di citazioni è diviso per anno e in base al motivo citazionale (*citation function*).

4.2.6 Ricerca in base alla venue di appartenenza degli articoli e ricerca in base al nome completo dell'autore

Nel dataset delle pubblicazioni di Springer sono presenti metadati di articoli di 126 tipi di `Journal` (entità della rivista tecnica) e di 81 tipi di `Book` (entità del libro dotato di ISBN, stampato in formato cartaceo o in formato elettronico). Per ricerca per venue si intende la ricerca di articoli sia pubblicati in `Journal`, quindi `JournalArticle`, sia in `Book`, quindi `BookChapter`. Essendoci più di una venue di pubblicazione, diversamente dal dataset di Semantic Lancet, è di buon senso poter affinare la ricerca in base alla venue, aspetto che sarà poi logico da approfondire in futuro.

Listing 4.3: Lista di venues (Journal e Book) per il dataset di Springer

```
SELECT DISTINCT ?journalTitle
```

```
WHERE {
                {
    ?journal a fabio:Journal;
        a fabio:Expression;
        dcterms:title ?journalTitle;
        datacite:hasIdentifier ?id.
    ?id a datacite:Identifier;
        datacite:usesIdentifierScheme datacite:occ;
        litre:hasLiteralValue ?literalValue.
                }
    union {
            ?journal a fabio:Book;
                dcterms:title ?journalTitle;
                datacite:hasIdentifier [
                    a datacite:Identifier;
                        datacite:usesIdentifierScheme datacite:isbn;
                        litre:hasLiteralValue ?bookIdVal;
                  ]
                }
}
```

Infine un'ultima piccola modifica riguarda il caso in cui un utente cerca gli articoli di un determinato autore: può scorrere a mano l'intera lista di autori trovati all'interno dei metadati del dataset, oppure può digitare ciò che si ricorda del nome dell'autore di cui vuole avere informazioni. In precedenza gli autori erano soltanto ricercabili per nome.

Conclusioni

Dato il fermento dei ricercatori nel campo del Semantic Publishing e la veloce diffusione della pubblicazione di scholarly Linked Open Data è ragionevole pensare di ampliare e mantenere un progetto che possa provvedere all'analisi avanzata di dati altrimenti interrogabili solo in modo diretto con query SPARQL. L'estensione di un progetto che provvede al sense-making degli scholarly Linked Open Data è ragionevole e apprezzata. Se non esistessero strumenti di questo tipo sarebbe difficile e dispendioso per i ricercatori universitari esplorare a fondo grandi dataset solo tramite interrogazione semantica. Rendere liberamente accessibili i Linked Data e progettare applicazioni finalizzate all'analisi dei dati non fanno che accelerare la ricerca scientifica. Fare di BEX un ambiente scalabile e flessibile tramite meccanismi di paginazione e implementando l'indipendenza da dataset con una struttura determinata permette di gestire un vasto numero di dati e usare l'applicazione con più scholarly Linked Open Dataset diversi.

Ci sono già molti sviluppi pensati per ampliare l'applicazione in futuro, e inoltre col passare del tempo si manifesteranno sicuramente nuove esigenze per le quali sarà opportuno fare ulteriori adattamenti, per esempio nel caso in cui cambi la struttura dei dataset o nel caso il cui l'interazione con l'utente faccia emergere la necessità di nuove funzionaità o variazioni di layout. L'applicazione ha alcuni limiti tra cui il problema di disambiguare i nomi degli autori a causa di come sono costruiti i datasets, il mancato aggiornamento costante dei datasets e problemi di lentezza di risposta in presenza di grande mole di dati. Inoltre lo SPARQL endpoint installato[12] non è ottimizzato per soddisfare grandi numeri di richieste, e l'esecuzione di queries e il caricamento dei risultati richiede spesso

[12] Apache Jena Fuseki v2

molto tempo, dunque si è pensato di migrare a un'altra piattaforma.

Integrazione con Open Citations

Uno tra gli sviluppi futuri è sicuramente l'unione di BEX con OCC (Open Citations Corpus)[28, 29], un dataset di metadati RDF completamente incentrato sulle bibliogafie degli articoli della letteratura accademica. Open Citations Corpus è parte di OpenCitations[13], un progetto attualmente in costruzione del Dipartimento di Informatica dell'Università di Bologna in collabolazione con l'Università di Oxford, e contiene materiale accurato su un grande numero di riferimenti bibliografici descritte nel dettaglio tramite ontologie SPAR. Il progetto OpenCitations si propone di creare uno scholarly Linked Open Dataset e di renderlo di dominio pubblico in modo che i contenuti possano essere sfruttati e riusati per qualsiasi scopo senza restrizioni legali.

Gestione e profilazione degli utenti

Un'altra idea per espandere BEX è la gestione e la profilazione degli utenti per personalizzare l'ambiente grafico tenendo conto delle preferenze di utilizzo in un'ottica di miglioramento della user-experience. Potrebbe anche essere interessante integrare un sistema di *recommendation* basato sul *collaborative filtering* cioè sulle valutazioni che l'utilizzatore dà agli oggetti presenti nel sistema[30].

Integrazione con motori semantici

Un interessante sviluppo futuro potrebbe essere l'utilizzo integrato di Apache Stanbol[14], un progetto dell' Apache Software Foundation per la gestione di contenuti semantici tramite motori semantici. Apache Stanbol automatizza processi di *content enhancement* cioè di identificazione ed estrazione di *named entities* (informazioni semantiche) in formato RDF dal contenuto di un testo, e attività di *reasoning* che riguardano l'aggiunta di ulteriori informazioni semantiche a quelle ottenute dal processo precedente. Lo stack di Apache Stanbol è modulare ed è formato componenti software riutilizzabili e accessibili tramite interfacce RESTful.

[13] http://opencitations.net/
[14] https://stanbol.apache.org/

Miglioramento di viste author-centric

In un'ottica author-centric è da tenere in considerazione l'idea di aggiungere la possibilità di scegliere una particolare venue di appartenenza nella vista di comparazione degli autori, affinchè si possa esplorare meglio il ruolo e l'influenza di un autore durante gli anni all'interno di una data rivista scientifica. L'impatto che un autore ha dipende anche dalla rivista in cui pubblica i propri articoli.

Miglioramento dell'efficienza e compatibilità con i browsers

In futuro sarà probabilmente necessario revisionare le prestazioni dell'applicazione in termini di velocità di caricamento dei dati e in termini di compatibilità con diversi browsers in modo da incrementare le performance e l'interattività.

Aggiunta di filtri

Si valuta infine l'implementazione di nuovi filtri per affinare la ricerca.

Bibliografia

[1] Di Iorio, A., Giannella, R., Poggi, F., Peroni, S., Vitali, F. (2015). *Exploring Scholarly Papers Through Citations* In Proceedings of the 2015 ACM Symposium on Document Engineering (DocEng '15). ACM, New York, NY, USA, 107-116.

[2] Giannella, R. (2015) *BEX: un ambiente user-friendly per esplorare articoli scientifici e bibliografie* Tesi di laurea, Scuola di Scienze, Corso di Laurea in Informatica per il Management, Alma Mater Studiorum Università di Bologna, 2013/2014. III.

[3] Peroni, S. (2014). *The Semantic Publishing and Referencing Ontologies* In Semantic Web Technologies and Legal Scholarly Publishing: 121-193. Springer, Cham, Switzerland

[4] Bagnacani, A., Ciancarini, P., Di Iorio, A., Nuzzolese, A. G., Peroni, S., Vitali, F. (2015). *The Semantic Lancet Project: a Linked Open Dataset for Scholarly Publishing* In Proceedings of Satellite Events of EKAW 2014, Lecture Notes in Artificial Intelligence 8982: 101-105 Berlin, Germany: Springer.

[5] Bagnacani, A., Ciancarini, P., Di Iorio, A., Nuzzolese, A. G., Peroni, S., Vitali, F. (2015). *Knowledge management of scholarly products: Semantic Lancet Project* Proceedings of 16th European Conference on Knowledge Management (ECKM 2015). Reading, UK: Academic Conferences and Publishing International.

[6] Berners-Lee, T., Hendler, J., Lassila, O. (2001). *The Semantic Web* Scientific american, 284(5), 28-37.

[7] Berners-Lee, T., Hendler, J., Lassila, O. (2006). *The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities* Scientific American

[8] Shotton, D., Portwin, K., Klyne, G., Miles, A. (2009). *Adventures in semantic publishing: exemplar semantic enhancements of a research article* PLoS Computational Biology, 5(4).

[9] De Waard, A. (2010). *From Proteins to Fairytales: Directions in Semantic Publishing* IEEE Intelligent Systems

[10] Berners-Lee, T., Hendler, J. (2001). *Publishing on the semantic web* Nature, 410(6832), 1023-1024.

[11] Yadagiri, N., Ramesh, P. (2013). *Semantic Web and the Libraries: An Overview* International Journal of Library Science, 7(1), 80-94.

[12] Peroni, S., Shotton, S., Vitali, F. (2012). *Scholarly publishing and Linked Data: describing roles, statuses, temporal and contextual extents* I-SEMANTICS, 8th Int. Conf. on Semantic Systems Sept. 5-7, 2012, Graz, Austria

[13] Berners-Lee, T. (2006). *Linked data* http://www.w3.org/DesignIssues/LinkedData.html

[14] Bizer, C., Heath, T., Berners-Lee, T. (2009). *Linked Data: The Story So Far* International Journal on Semantic Web and Information Systems 5 (3): 1-22.

[15] Barnaghi, P., Presser, M., Moessner, K. *Publishing Linked Sensor Data* IOS Press

[16] Kruk, S. R., McDaniel, B. (2009). *Semantic Digital Libraries* Springer

[17] Kruk, S. R., Woroniecki, T., Gzella, A., Dabrowski, M. (2007). *JeromeDL: a Semantic Digital Library*

[18] Baruzzo, A., Casoto, P., Challapalli, P., Dattolo, A., Pudota, N., Tasso, C. (2009).*Toward Semantic Digital Libraries: Exploiting Web2.0 and Semantic Services in Cultural Heritage*

[19] Nuzzolese, A. G., Gentile, A. L., Presutti, V., Gangemi, A. (2016). *Semantic web conference ontology - a refactoring solution* In The Semantic Web: ESWC 2016 - Satellite Events (to appear). Springer, 2016.

[20] Gentile, A. L., Acosta, M., Costabello, L., Nuzzolese, A. G., Presutti, V., Recupero, D. R. (2015). *Conference Live: Accessible and Sociable Conference Semantic Data* In Proceedings of WWW 2015 (Companion Volume), pages 1007-1012. ACM. DOI: 10.1145/2740908.2742025

[21] Gentile, A. L., Nuzzolese, A. G. (2015). *cLODg - Conference Linked Open Data Generator* In S. Villata, J. Z. Pan, and M. Dragoni, editors, Proceedings of the ISWC 2015 Posters & Demonstrations Track co-located with the 14th International Semantic Web Conference (ISWC-2015), Bethlehem, PA, USA, October 11, 2015., volume 1486 of CEUR Workshop Proceedings. CEUR-WS.org, 2015.

[22] Osborne, F., Motta, E., Mulholland, P. (2013). *Exploring Scholarly Data with Rexplore* International Semantic Web Conference, Sydney, Australia

[23] Li, H., Councill, I., Lee, W-C., Giles, L. (2006). *CiteSeerx: an architecture and web service design for an academic document search engine*

[24] Valenzuela, M., Ha, V., Etzioni, O. (2015).*Identifying Meaningful Citations* AAAI Workshop on Scholarly Big Data

[25] Niyazov, Y., Vogel, K., Price, R., Lund, B., Judd, D., Akil, A., Mortonson, M., Schwartzman, J., Shron M. (2016).*Open Access Meets Discoverability: Citations to Articles Posted to Academia.edu* PloS one, 11(2), e0148257.

[26] Shneiderman, B. (1996). *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations* VL.

[27] Falco, R., Gangemi, A., Peroni, S., Vitali, F. (2014). *Modelling OWL ontologies with Graffoo* ESWC 2014 Satellite Events - Revised Selected Papers, Lecture Notes in Computer Science 8798: 320-325. Berlin, Germany: Springer.

[28] Peroni, S., Dutton, A., Gray, T., Shotton, D. (2015). *Setting our bibliographic references free: towards open citation data.* Journal of Documentation, 71 (2): 253-277.

[29] Shotton, D. (2013). *Open citations* Nature, 502 (7471).

[30] Borgiani, S. (2016). *Progettazione ed implementazione di un recommendation system di articoli scientifici basato su Apache Mahout* Tesi di laurea, Scuola di Scienze, Corso di Laurea in Informatica per il Management, Alma Mater Studiorum Università di Bologna, 2014/2015. III.

Ringraziamenti

Un ringraziamento particolare è riservato al Prof. Angelo Di Iorio per l'interesse che ha dimostrato di avere verso il lavoro dei suoi studenti, e per la disponibilità che ha avuto nei miei confronti.

Ringrazio lo sviluppatore di BEX 1.0 'Lele' Giannella per le dritte che mi ha dato, spero che ci saluteremo di persona un giorno!

Grazie a tutta la mia famiglia.

Grazie infinite alle persone su cui posso contare davvero: le mie developers Silvia e Silvia, Lea, anche se ormai viviamo in continenti diversi riesce sempre a essere presente, e le mie amiche di sempre Cecilia e Federica.

Grazie agli amici dell'università Antonio, Riccardo, Gabriele e Simone, e alle amichette di Bologna Federica, Ilaria, Lara e Pamela.