Il Portale

Bibliografiapiste

Stefano Zamblera

1 Risorse bibliografiche digitali

1.1 Biblioteche digitali

La nozione astratta di *biblioteca digitale*[1] concerne la rappresentazione digitale del contenuto informativo di una biblioteca e delle *metainformazioni* (o *metadati*) atte al reperimento di specifiche sezioni al suo interno.

Tale contenuto ha la forma di un insieme di documenti dotato di un'organizzazione complessiva dovuta ad un agente intenzionale distinto dai creatori dei singoli documenti in essa contenuti.

La nozione di "*sistema di biblioteca digitale*", invece, attiene alle risorse tecnologiche (risorse *hardware*, *sistemi di rete*, *software* per l'archiviazione dei dati, interfacce utente e *sistemi* di *information retrival*) necessarie ad implementare tale modello, e di conseguenze individua le funzioni e i servizi che vengono messi a disposizione degli utenti.

Alla luce di queste riflessioni definiamo "*biblioteca digitale*" *una collezione di documenti digitali strutturati* (sia prodotti mediante *digitalizzazione* di originali materiali, sia realizzati ex-novo), dotata di un'organizzazione complessiva coerente di natura semantica e tematica, che si manifesta mediante un insieme di relazioni interdocumentali e intradocumentali e mediante un adeguato apparato metainformativo.

In questo senso possiamo distinguere una *biblioteca digitale* da un insieme non organizzato di informazioni assolutamente eterogenee come *World Wide Web*, ma anche da molti archivi testuali che attualmente sono disponibili su Internet e che si presentano come 'depositi testuali piuttosto che come vere e proprie biblioteche.

---

[1] I primi spunti in questo campo precedono la nascita di Internet e persino lo sviluppo dei computer digitali (articolo di Vannevar Bush, As we May Think, 1945, traduzione italiana in T. Nelson, Literary Machine 90.1, 1992, dove il tecnologo americano immagina l'ormai celeberrimo *Memex*).
Si trattava di una sorta di scrivania automatizzata, dotata di un sistema di proiezione di microfilm e di una serie di apparati che consentivano di collegare tra loro i documenti su di essi fotografati.
Lo stesso Bush, V. Bush, Op. Cit., par. 1/38, introducendo la descrizione del suo ingegnoso sistema di ricerca e consultazione di documenti interrelati, lo definì una "sorta di archivio e biblioteca privati".
Una approssimazione maggiore all'idea di biblioteca digitale (sebbene il termine non compaia esplicitamente), si ritrova nel concetto elaborato da Ted Nelson, *Literary Machine* 90.1: il progetto *Xanadu*, Muzzio, Padova 1992, cui dobbiamo anche la prima formulazione esplicita dell'idea di ipertesto digitale. Nelson, sin dai suoi primi scritti degli anni 60, descrive un sistema ipertestuale distribuito (che poi battezzerà *Xanadu*) costituito da una rete di documenti e dotato di un sistema di indirizzamento e di reperimento. La convergenza teorica e tecnica tra biblioteche digitali e sistemi ipertestuali distribuiti trova infine pieno compimento con la nascita e lo sviluppo di World Wide Web. L'ambiente ipertestuale della rete Internet, infatti, ha fornito un ambiente ideale per la sperimentazione concreta e diffusa di tutta l'elaborazione teorica accumulata in questo settore negli anni passati.

Una delle metafore ricorrenti per descrivere il fenomeno Internet è infatti quella di una biblioteca: come una biblioteca, la rete contiene una quantità enorme di documenti testuali (e non testuali), ha i suoi cataloghi, i suoi strumenti di ricerca dell'informazione.

Soprattutto, a differenza di ogni biblioteca reale del mondo, Internet sembra non avere limiti nella capacità di contenere e diffondere informazioni e sembra anzi realizzare, per mezzo della tecnologia, il mito della biblioteca universale, che accompagna l'umanità da molti secoli.[2]

In realtà il parallelo metaforico tra la rete e il concetto di biblioteca universale è in parte fuorviante.

Lo spazio informativo della rete, e in particolare quello del web, non è uno spazio completamente strutturato, al contrario esso per alcuni suoi applicativi ha teso alla "*non organizzazione*" in virtù della sua estrema dinamicità e fluidità.

I vari strumenti di ricerca delle informazioni in rete dunque non rendono conto della totalità dei contenuti informativi presenti sulla rete stessa: essi ne tracciano semmai mappe parziali e locali.

Lo spazio informativo di una biblioteca invece deve essere uno spazio completamente strutturato e organizzato che trova una rappresentazione esaustiva nei vari tipi di cataloghi di cui essa ha la necessità di essere dotata per poter rendere fruibili i propri contenuti.

Nondimeno, sulla rete non mancano servizi informativi strutturati, e tra questi, sebbene sembri un gioco di parole, spiccano proprio i servizi gestiti dalle *biblioteche* "*reali*".

L'incontro tra Internet e biblioteche, che ha ormai una storia assai lunga, è stato fortemente propiziato dal radicamento della rete nel mondo universitario statunitense.

Gli Stati Uniti, infatti, hanno un enorme patrimonio di biblioteche,[3] tra cui spiccano le biblioteche universitarie, tradizionalmente dotate di servizi al pubblico assai avanzati ed efficienti.

La predisposizione di servizi on-line da parte di queste istituzioni è stata, nella gran parte dei casi, un'evoluzione naturale, ma, in generale, si deve rilevare che il fenomeno Internet ha suscitato nel mondo bibliotecario un vasto interesse anche al di fuori degli Stati Uniti.

---

[2] In effetti, sin dalle origini, la biblioteca è stata concepita come uno strumento di conservazione universale del sapere, in cui fosse consentito a chiunque un immediato accesso alla conoscenza depositata nei documenti. James O'Donnell cita ad esempio la *"Lettera di Aristeo a Philocrate"*, in cui l'autore parlando della biblioteca di Alessandria, attribuisce a Demetrio di Phaleron, ministro della cultura del faraone Tolomeo, l'intenzione di raccogliere nella sua meravigliosa collezione tutti libri del mondo. Al mito della biblioteca universale è stato dedicato il convegno The universal library: From Alexandria to the Internet, (URL: http://www.fdgroup.co.uk/univweb.htm), organizzato nel settembre del 1997 dal Library History Group della Library Association (URL: http://www.fdgroup.co.uk/lhg.htm).

[3] In base a stime recenti si contano oltre 120 mila biblioteche, di cui 3 mila e cinquecento a carattere universitario.

In virtù di tale interesse, Internet offre oggi una notevole quantità di servizi di tipo bibliotecario rivolti al pubblico generico, oltre ad alcuni servizi orientati maggiormente ad una utenza professionale.

Possiamo suddividere tale insieme di servizi nelle seguenti classi:

- servizi di informazione al pubblico basati sul web relativi a singole biblioteche (*information desk on-line*),
- servizi di consultazione on-line dei cataloghi informatici di singole biblioteche o di gruppi di biblioteche (*cataloghi individuali e collettivi*),
- servizi di distribuzione selettiva di documenti (*document delivery*),
- servizi speciali di informazione e di supporto per i bibliotecari,
- servizi di *biblioteca digitale*.

Il primo tipo di servizi è costituito dai siti web approntati da singole biblioteche che offrono al pubblico informazioni, a vario livello di dettaglio, sulla biblioteca stessa, sulla sua collocazione, sui regolamenti e gli orari di accesso, sulla qualità e consistenza delle collezioni.

In alcuni casi è possibile trovare anche servizi avanzati come la prenotazione del prestito di un volume, o persino l'attivazione di procedure per il prestito interbibliotecario (di norma questi servizi sono approntati da biblioteche universitarie, ed hanno un accesso limitato).

Naturalmente la disponibilità di questi ultimi strumenti è legata alla presenza sul sito bibliotecario di un sistema di consultazione on-line del catalogo.

Tali sistemi, detti *OPAC* (*acronimo* di *Online Public Access Catalog*), sono senza dubbio una delle più preziose risorse informative attualmente disponibili sulla rete.

Essi sono il prodotto di una lunga fase di innovazione tecnologica all'interno delle biblioteche che ha avuto inizio sin dagli anni sessanta, e che ha avuto tempi di espletamento e di diffusione capillare assai differenziati.

Tutt'oggi, solo in alcuni casi l'automazione bibliotecaria è arrivata a pieno compimento, portando alla sostituzione totale dello schedario cartaceo con sistemi informatici.[4]

[4] E peraltro si rileva una notevole sperequazione nell'adozione di sistemi informatici nelle biblioteche sia a livello internazionale sia all'interno degli ambiti nazionali.

L'automazione dei sistemi catalografici si è incontrata ben presto con lo sviluppo delle tecnologie telematiche, ed in particolare con la diffusione della rete Internet nell'ambito del circuito accademico.

Il passaggio dal catalogo informatizzato al catalogo on-line, infatti, ha comportato una evoluzione lineare, che si è verificata in un contesto già informatizzato e dunque non restio all'innovazione.

Attualmente le biblioteche, di maggiori o minori dimensioni, universitarie, pubbliche e private, che, oltre ad avere un loro sito su Internet, danno agli utenti la possibilità di consultare on-line i cataloghi delle loro collezioni, sono nell'ordine delle migliaia.[5]

Se la possibilità di effettuare ricerche bibliografiche in rete è ormai un dato acquisito, diverso è il discorso per quanto riguarda l'accesso diretto ai documenti: infatti, il passaggio dalla biblioteca informatizzata alla biblioteca digitale è appena agli inizi.

Intendendo con il termine *biblioteca digitale*, in prima approssimazione, *un servizio on-line che produce, organizza e distribuisce sulla rete*, in vario modo, versioni digitali di documenti e testi, ad un livello intermedio vediamo collocarsi i servizi di distribuzione selettiva dei documenti (document delivery).

A questa categoria appartengono organizzazioni ed enti che archiviano e spogliano grandi quantità di pubblicazioni periodiche e che permettono a studiosi o ad altri enti bibliotecari di acquistare singoli articoli, che vengono poi spediti via posta, fax o e-mail, una risorsa preziosa per chi deve effettuare attività di ricerca e non ha a disposizione una biblioteca dotata di una collezione di periodici sufficientemente esaustiva.

Le prime pionieristiche sperimentazioni nel campo delle biblioteche digitali, come vedremo, sono quasi coeve alla nascita di Internet.

Ma è soprattutto dall'inizio di questo decennio che si è assistito ad una notevole crescita delle sperimentazioni e dei progetti, alcuni dei quali finanziati da grandi enti pubblici in vari paesi. Parallelamente alla proliferazione di iniziative, si è avuta una crescente attenzione teorica e metodologica al tema delle biblioteche digitali, tanto da giustificare la sedimentazione di un dominio disciplinare autonomo.

Alla costituzione di questo dominio hanno fornito importanti contributi vari settori della ricerca informatica e sui nuovi media, come l'area del *text processing*, dell'*information retrival* e degli *agenti software*, della *grafica computerizzata*, della *telematica* e delle *reti*, ma senza dubbio i contributi di maggiore rilievo sono venuti dalle ricerche sui sistemi informativi distribuiti e dalla teoria degli ipertesti, nel cui contesto si può rintracciare la genealogia stessa dell'idea di "*biblioteca digitale*".

Da un punto di vista generale una biblioteca si può definire *digitale* o *elettronica* almeno in due distinti casi:

[5] Vedi nota 3.

1. quando è una raccolta di testi veicolata da un supporto per la cui produzione o diffusione si ricorre all'uso delle cosiddette moderne tecnologie, intendendo con ciò non soltanto gli "scaffali" di internet ma anche un qualsiasi archivio di dati testuali riprodotto su disco ottico, purché abbia la caratteristica di essere rappresentativo di una raccolta di opere,

2. quando mette a disposizione dei suoi utenti alcuni servizi bibliografici di tipo tradizionale, come per esempio la consultazione del catalogo, ma utilizza il canale della rete e la modalità elettronica per la sua lettura.

Da un punto di vista teorico, i due termini, non sono equivalenti: *digitale* non è sinonimo di *elettronico*, anche se nell'insieme dei termini con cui si è soliti etichettare al loro apparire fenomeni del tutto nuovi capita, sovente di vederli utilizzati indifferentemente.

Prescindendo dalle distinzioni terminologiche, in entrambi i casi siamo di fronte ad una versione *altra* di un indice alfabetico (*catalogo*) e/o di un documento (*testo*) che ha il suo antecedente diretto nella storia e nella realtà delle biblioteche e dei loro servizi forniti da archivi e supporti cartacei.

Attualmente nella rete *internet* è contemplata la presenza indistinta di entrambe le tipologie, talvolta con una coincidenza dei due aspetti in un'unica fattispecie: la stessa biblioteca infatti può aprire le sue porte al pubblico della rete per consultare il catalogo in linea e contemporaneamente per *sfogliare* direttamente al monitor una parte delle proprie edizioni, riprodotte da esemplari cartacei.

In altri casi, invece, può presentarsi una netta separazione dei due aspetti: ciò avviene tutte le volte che una biblioteca decide, per esempio, di voler rendere disponibile in rete soltanto le notizie più strettamente legate ai servizi di informazione bibliografica.

Oppure, in altri casi ancora, si può trattare di una *biblioteca digitale* che, sfruttando l'ubiquità dell'architettura distribuita delle informazioni propria del web, si chiama pur sempre biblioteca, ma soltanto in senso metaforico, dato che non si ricongiunge nominalmente a nessuna preesistente istituzione, pur preservandone idealmente alcune delle finalità di documentazione e conservazione della cultura scritta e di diffusione della lettura.

Sempre più spesso si assiste infatti in *internet* al sorgere di iniziative di editoria elettronica e di conseguente diffusione digitale dei testi così riprodotti, che sono il frutto del lavoro di un gruppo più o meno anonimo di volontari.

In tutti i casi, sia che si tratti di una biblioteca senza libri, o che sia invece una biblioteca con libri di carta e libri di *bit*, più che di vera e propria biblioteca sarebbe più corretto forse parlare di *metabiblioteca*, sottolineando con ciò

soprattutto la totale assenza di confini spaziali delle sue raccolte, quale elemento veramente distintivo ed innovativo rispetto all'archetipo storico.

Ma, come anticipato precedentemente, se molta è la confusione sotto il cielo delle biblioteche digitali, non si può dire che le cose vadano tanto meglio in quello dei loro oggetti privilegiati: i cosiddetti testi elettronici.

Edizioni elettroniche di documenti nati solo per la rete si affiancano a riproduzioni digitali di edizioni di testi a stampa, quasi fossero le une rispetto alle altre edizioni di seconda generazione, che ne conservano inalterate, nonostante la *digitalizzazione*, le caratteristiche tipografiche.

Se il termine *edizione* non è più legato unicamente allo strumento tipografico, ma anche al concetto di manipolazione e di diffusione del testo scritto, allora forse è giusto chiedersi come debba definirsi per esempio la riproduzione digitale di un manoscritto o di un codice che attraverso la presentazione ipertestuale permette la lettura dei vari livelli stratificati di scrittura.[6]

In ogni caso, i siti di carattere bibliotecario accessibili attraverso Internet sono ormai migliaia, ed è ovviamente impossibile elencarli tutti.

Come sempre, però, la rete fornisce ai suoi utenti degli strumenti di orientamento di secondo livello.

Esistono infatti diversi *repertori* di siti bibliotecari, che possono essere consultati per scoprire l'*URL* della biblioteca che si sta cercando (ammesso che ne abbia uno), o per individuare quali biblioteche in una certa area geografica sono dotati di servizi in rete.

Occorre tuttavia ricordare che non tutte le biblioteche dotate di un sito web hanno anche un *OPAC* pubblico, o (evenienza più rara) che alcuni *OPAC* non sono associati ad un sito web.

Purtroppo i repertori di siti bibliotecari non sempre tengono nel dovuto conto queste distinzioni, specialmente se non sono specificamente dedicati alle risorse bibliotecarie.

Rientrano in questa categoria tutti i repertori di siti bibliotecari che fanno parte di più vasti repertori di risorse di rete, come quello organizzato da *Yahoo*[7] o da *Excite*.[8]

Passando ai repertori specializzati in siti bibliotecari, uno dei più aggiornati e completi è Libweb realizzato alla *University of Berkeley*, in California, a cura di Thomas Dowling.[9]

---

[6] In attesa che il rigore scientifico derivante dalla definizione di *standard* universali di descrizione catalografica sia applicato a questo tipo di risorse e porti ad una prassi consolidata che serva a chiarire un panorama così vario, (magari identificando in modo univoco gli oggetti rappresentati e la loro indicizzazione a volte lasciata in mano all'esclusivo appannaggio dei *software* di indicizzazione dei documenti e del recupero dell'informazione).

[7] http://www.yahoo.com/Reference/Libraries

[8] http://www.excite.com/education/libraries

[9] http://sunsite.berkeley.edu/Libweb : l'elenco è diviso per aree geografiche (Stati Uniti, Africa, Asia, Australia, Europa, Sud America, Canada), e successivamente per nazioni. Solo il

Un altro ottimo repertorio globale di *OPAC* basati sul *web* è *webCats*: in questo caso l'elenco può essere scorso in base a tre criteri di ordinamento: per aree geografiche e nazioni, per tipologia di biblioteca e per tipo di *software*.

Quest'ultima categoria articola i vari *OPAC* in base al prodotto di catalogazione utilizzato, e può essere utile per coloro che hanno dimestichezza con l'interfaccia e la sintassi di ricerca di uno di essi.[10]

Pur se con un certo ritardo, oramai sono molte le biblioteche italiane che hanno realizzato dei sistemi *OPAC* su Internet.

Il migliore repertorio di *OPAC* italiani è ospitato sull'ottimo sito web della *Associazione Italiana Biblioteche*.[11]

Il repertorio è suddiviso in due sezioni: una dedicata ai cataloghi collettivi nazionali, e una sezione dedicata ai cataloghi collettivi regionali, provinciali, comunali e ai cataloghi di singole biblioteche.

Per ciascun *OPAC* vengono forniti delle brevi note informative e una serie di link alle pagine di ricerca e alle eventuali pagine di istruzioni per l'uso.

Oltre al repertorio, l'*AIB*, in collaborazione con il *CILEA*, ha realizzato in via sperimentale il *Meta-OPAC Azalai Italiano* (*MAI*).

Si tratta di un sistema di interrogazione unificato dei cataloghi bibliotecari italiani su Internet, che permette di inviare una medesima ricerca a più *OPAC* contemporaneamente.

MAI permette di selezionare in anticipo quali cataloghi interrogare (in base alla collocazione geografica o al tipo di biblioteca), e poi fornisce una

---

ramo dedicato alle biblioteche statunitensi è articolato anche per tipo di biblioteca. Oltre alla possibilità di scorrere il repertorio, Libweb fornisce anche un sistema di ricerca per parole chiave, basato su una sintassi abbastanza semplice.
Molto completo è anche il repertorio Bibliotheks-OPACs und Informationsseiten, http://www.hbz-nrw.de/hbz/toolbox/opac.htm, curato da Hans-Dieter Hartges. Si tratta di una unica grande pagina web che elenca centinaia di servizi OPAC con interfaccia web, classificandoli per nazioni.

[10] Per quanto concerne webCats il portale è curato curato da Peter Scott, ed il suo URL è: http://www.lights.com/webcats/.
Sempre a Peter Scott si deve il repertorio di OPAC basato su interfaccia a caratteri denominato Hytelnet. In origine Hytelnet era un programma indipendente, anche esso basato su interfaccia a caratteri (ne esistevano varie versioni), che permetteva di navigare attraverso un repertorio di OPAC strutturato per aree geografiche. La sua interfaccia era basata su una serie di menu gerarchici attraverso i quali si poteva 'scenderè all'indirizzo della singola risorsa. Successivamente Scott ha realizzato un gateway tra Hytelnet e web, il cui aggiornamento è stato sospeso nel 1997. Tuttavia, poiché si tratta di un repertorio di OPAC basati su telnet, può ancora essere di grande utilità. Il sito web ufficiale di questo servizio è http://www.lights.com/hytelnet. Una volta selezionata la biblioteca che si desidera consultare si arriva ad una pagina web in cui, oltre al link diretto con l'OPAC (che naturalmente avvia una sessione telnet), sono contenute le istruzioni per effettuare la procedura di accesso e un link che porta ad un breve manuale sulla sintassi di ricerca dei principali software di catalogazione informatica.

[11] *AIB*, http://www.aib.it , mentre l'URL del repertorio è http://www.aib.it/aib/lis/opac1.htm

maschera in cui è possibile specificare i termini di ricerca (ovviamente occorre tenere conto che non tutte le chiavi di ricerca sono disponibili su tutti i sistemi).

Il risultato dell'interrogazione viene composto in una unica pagina web che mostra l'output di ciascun catalogo, completo di pulsanti e collegamenti per visualizzare la scheda bibliografica o per raffinare la ricerca.

Un altro repertorio di siti bibliotecari italiani (anche se non necessariamente di cataloghi on-line) è *Biblioteche italiane*,[12] a cura del *Sistema bibliotecario del Politecnico di Torino*, anche esso organizzato per aree geografiche.[13]

[12] http://www.biblio.polito.it/it/documentazione/biblioit.html

[13] Per un repertorio dettagliato dei siti a carattere bibliotecario, e per una lista di *OPAC italiani* e *stranieri* è possibile consultare *l'ebook* pubblicato da *Laterza.it* reperibile all'URL: http://www.laterza.it/internet/leggi/internet2000/online/testo/23_testo.htm

1.2 utilizzo delle risorse bibliografiche online per l'Egittologia

Sono oramai presenti diverse risorse bibliografiche digitali, che permettono consultazioni più o meno approfondite e personalizzabili tramite ricerche per parole chiave e che costituiscono un'ottima risorsa nel campo egittologico: ad esempio le risorse realizzate dall'*Oriental Institute* dell'*Università* di *Chicago*,[14] o dell'*Università* di *Leiden*,[15] sono strumenti potenti ed aggiornati e forniscono un catalogo di testi con elenchi per argomento, per autore, per collezione, ecc…

Le risorse elettroniche proprie dell'Oriental Institute di Chicago costituiscono probabilmente l'archivio digitale egittologico più completo e ricco di applicativi presenti oggigiorno sul web.

È utile sommarizzare tramite questa scheda tutte le risorse altrimenti lunghissime da descrivere, presenti sul sito dell'*Oriental Institute di Chicago*.

Per approfondirne le caratteristiche e le specifiche rimandiamo ai link citati, sufficienti per raggiungere tutte le risorse riportate:

<table>
<tr><td>Ente :</td><td>The Oriental Institute of The University of Chicago</td></tr>
<tr><td>URL :</td><td>http://www-oi.uchicago.edu/OI/default.html</td></tr>
<tr><td>Risorse :</td><td>

On-Line Research Archives of the Oriental Institute, University of Chicago,
URL: http://oilib.uchicago.edu/oilibcat.html con criteri di ricerca:
<table>
<tr><td>KEYWORD</td><td>- Find items that contain specific words</td></tr>
<tr><td>AUTHOR</td><td>- Find items by author's name</td></tr>
<tr><td>TITLE</td><td>- Find items by their title</td></tr>
<tr><td>SUBJECT</td><td>- Find titles by subject</td></tr>
<tr><td>LCCN or ISBN</td><td>- Find titles by LCCN or ISBN</td></tr>
<tr><td>EXPERT</td><td>- Expert searching</td></tr>
<tr><td>Acquisitions</td><td>- It is possible to receive the lists of the Acquisitions of Research Archives by e-mail</td></tr>
</table>

ORIENTAL INSTITUTE ELECTRONIC PUBLICATIONS ON-LINE,
URL: http://www-oi.uchicago.edu/OI/OI_Electronic_Resources.html

The Demotic Dictionary of the Oriental Institute of the University of Chicago, Janet H. Johnson, editor
Thus Wrote 'Onchsheshonqy - An Introductory Grammar of Demotic (Third

</td></tr>
</table>

[14] http://www.etana.org/abzu
[15] http://www.leidenuniv.nl/nino/aeb.html

	Edition), Janet H. Johnson • RECOMMENDED READING ON THE ANCIENT NEAR EAST • ANE - Electronic Discussion List • ORIENTAL INSTITUTE MUSEUM: THE VIRTUAL MUSEUM • ORIENTAL INSTITUTE MUSEUM: HIGHLIGHTS FROM THE COLLECTIONS • PHOTOGRAPHIC ARCHIVES o EGYPT, IRAN, IRAQ, and SUDAN (66 Photographs) o PERSEPOLIS and ANCIENT IRAN (967 Photographs) o THE 1905-1907 BREASTED EXPEDITIONS to EGYPT and the SUDAN (1055 Photographs) • ACHAEMENID ROYAL INSCRIPTIONS PROJECT • XSTAR - XML System for Textual and Archaeological Research • TOM VAN EYNDE: THEBES PHOTOGRAPHIC PROJECT • Treasures from the ROYAL TOMBS of UR Exhibition • ORIENTAL INSTITUTE MAP SERIES • ARCHAEOLOGICAL SITE PHOTOGRAPHY Mesopotamia (414 photos) and Egypt (579 photos) • SUPPORTERS OF ORIENTAL INSTITUTE ELECTRONIC RESOURCES • ANNUAL ELECTRONIC RESAORCES o 2000-2001 Annual Report on Electronic Resources o 1999-2000 Annual Report on Electronic Resources o 1997-98 Annual Report on Electronic Resources o 1996-97 Annual Report on Electronic Resources o 1995-96 Annual Report on Electronic Resources o 1994-95 Annual Report on Electronic Resources

Anche il progetto *ABZU*, risultato della collaborazione di molti enti,[16] fra i quali è nuovamente da citare l'*Oriental Institute* di *Chicago*, costituisce una

[16] ABZU è un progetto della ETANA nato dalla collaborazione di: The American Schools of Oriental Research, Case Western Reserve University, Cobb Institute of Archaeology at Mississippi State, Oriental Institute of the University of Chicago, Society of Biblical Literature, Sonia and Marco Nadler Institute of Archaeology of Tel Aviv University, Vanderbilt University.

eccellente fonte di documentazione egittologia, ed anche per essa è utile riassumerne le risorse:

Ente :	ABZU, Oriental Institute at the University of Chicago
URL :	http://www.etana.org/abzu/
Risorse :	Criteri di ricerca: • AUTHOR • TITLE • SUBJECT • JOURNAL TITLE • KEYWORD ANYWHERE Limiti del dominio di ricerca: • LIMIT TO ARTICLES • LIMIT TO BOOKS • LIMIT TO BOOKS CHAPTERS • LIMIT TO PRSENTATIONS • LIMIT TO WEB SITES • LIMIT TO LIMIT TO DATES Risultati ordinabili: • SORT BY AUTHOR • SORT BY DATE • SORT BY TITLE

Tutte queste risorse consentono una consultazione veloce dei testi archiviati, e permettono così al ricercatore di ottenere una lista bibliografica inerente agli argomenti di suo interesse aggiornatissima e completa.

La risorsa fornita dal sito della *Leiden University* riporta invece la raccolta bibliografica egittologia annuale,[17] ed organizza i propri corpora testuali secondo indici annuali:

Ente :	Netherlands Institute for the Near East, Leiden University
URL :	http://www.leidenuniv.nl/nino/nino.html
Risorse :	*AEB, Annual Egyptological Bibliography*, URL: http://www.leidenuniv.nl/nino/aeb.html • About the AEB • AEB 1992 • AEB 1993 • AEB 1994 • AEB 1995 • AEB 1996 • AEB 1998 • Books and articles: 1999-2002 • Subscriptions / orders

[17] *Annual Egyptological Bibliography, AEB*

L'idea di pubblicare la bibliografia relativa alle *piste carovaniere del deserto occidentale egiziano* è nata proprio nell'ordine di fornire un servizio dedicato alla ricerca egittologia, e contenesse tutte le caratteristiche presenti nelle risorse elettroniche appena citate, avvalendosi però di tutte le migliori tecnologie informatiche per l'amministrazione ed il recupero delle informazioni.

La risorsa da realizzare, inerente al tema, indubbiamente peculiare anche all'interno della specializzazione egittologica stessa, avrebbe dovuto fornire due principali risorse fondamentali per la ricerca nel settore:

- *bibliografia topografica*,
- *catalogo OPAC*.

La *bibliografia topografica* avrebbe avuto il duplice compito di raggruppare ed organizzare secondo un indice topografico tutta la bibliografia egittologia reperibile sulle *piste carovaniere del deserto occidentale egiziano*, partendo e basandosi sulla struttura realizzata da *Bertha Porter* e *Rosalind Moss*, *Topographical Bibliography of Ancient Egyptian Hieroglyphic Texts*, naturalmente applicata all'argomento specifico ed aggiornandone i contenuti, dato il semplice fatto che la bibliografia topografica cartacea redatta dalle due autrici contemplasse quanto pubblicato e scoperto sino al 1943.

Questo tipo di strumento avrebbe perfettamente aderito alle caratteristiche dell'argomento studiato, in quanto quest'ultimo richiedesse sostanzialmente chiarezza e schematicità per la migliore comprensione della rete di scambi e delle vie di collegamento sviluppatesi nel deserto occidentale egiziano nel corso della storia antica.

La bibliografia topografica, dunque, sarebbe stata organizzata e citata in base alle singole voci dell'indice topografico.

Sarebbe poi stata necessaria una ulteriore organizzazione del materiale bibliografico, capace di rappresentare non le citazioni specifiche dei documenti reperiti, ma rendendo fruibili i singoli testi bibliografici secondo uno schedario ed un catalogo canonico.

La risorsa online sarebbe dunque stata organizzata anche in un catalogo elettronico di tipo *OPAC*, offrendo così all'utente uno strumento di ricerca più elastico per un ulteriore approfondimento ed una corretta consultazione della bibliografia inerente al tema specifico: *le piste carovaniere del deserto occidentale*.

2 Reperimento delle informazioni

2.1 MultiLingual Information Access

La rapida diffusione del *World Wide Web* ha consentito un'enorme distribuzione delle risorse di comunicazione, disseminando una grande mole di documenti per tutto il mondo.

Questa situazione ha visto l'espansione parallela dei linguaggi rappresentati a quella verificatasi per l'utenza di *Internet*, situazione che avrebbe richiesto di ovviare all'impostazione iniziale ormai inaccettabile di reperibilità delle informazioni basata su standard di lingua inglese.[18]

La rapida e più recente diffusione della tecnologia del *World Wide Web* ha dunque prodotto una espansione enorme del numero di pagine redatte e di utenti parlanti lingua non inglese.[19]

Proprio sulla base di questi dati statistici sono state sviluppate tutte quelle strutture necessarie all'implementazione di sistemi funzionanti per il reperimento delle informazioni presenti sul web a prescindere dal linguaggio di redazione ed indipendentemente dalla lingua parlata dall'utente.[20]

---

[18] Basti pensare ai primi motori di ricerca: *Yahoo*, *AltaVista* e *Lycos*, ec… e questo pechè la diffusione del *web* sarebbe inizialmente avvenuta in ambienti accademici o comunque sufficientemente istruiti per una comprensione soddisfacente della lingua inglese.

[19] *First DELOS International Summer School on Digital Library Technologies*, Pisa, 9-13 luglio 2001:
Nel suo intervento, *Carol Peters* (*IEI-CNR*) ha descritto lo stato delle ricerche sullo sviluppo dei sistemi e delle tecnologie per il *MultiLingual Information Access* (*M.L.I.A.*), che consente l'archiviazione, il recupero e l'accesso a informazioni in tutte le lingue, e del *Cross Language Information Retrival* (C.L.I.R.), che permette di svolgere una ricerca in una sola lingua e poter recuperare documenti in varie lingue.
Se infatti fino ad oggi *Internet* è stato dominato dalla lingua inglese, già da tempo si assiste ad un aumento significativo di documenti in lingue diverse; di conseguenza cresce la richiesta, da parte delle comunità di utenti non anglofone, che siano sviluppati quei sistemi che favoriscono l'accesso all'informazione a prescindere da ogni barriera linguistica e culturale.
Tutto questo ha un impatto molto forte sulle nostre attività, in particolare nel campo della formazione, dell'e-commerce e del divertimento, soprattutto dal momento che *Internet* viene sempre di più utilizzato non solo dal mondo accademico, ma da un'utenza generalizzata
Molti altri aspetti che caratterizzano la costruzione delle *digital libraries* sono stati trattati durante il corso: *Norbert Fuhr* (*Università di Dortmund*) ha spiegato l'applicazione dell'information retrival al mondo delle biblioteche digitali, sottolineando l'importanza di modelli concettuali adeguati (per esempio gli *FRBR* dello *IFLA*) in grado di descrivere i tipi di oggetti e le relazioni esistenti tra loro.
Andreas Paepke (*Stanford University*) ha descritto le problematiche relative all'elaborazione di un'interfaccia che sia sempre più semplice per l'utente ma contemporaneamente dotata di una struttura sempre più forte e funzionale.
Ha portato l'esempio della tecnologia dei microcomputer che, a causa delle ridotte dimensioni, rende necessario un'attenzione particolare alle interfacce.

[20] Un sistema di ricerca delle informazioni, *Information Retrival System, I.R.S.*, è un corredo di strumenti che consentono di esplorare una collezione di documenti.

Il termine *MultiLingual Information Access* utilizzato nelle sue accezioni più ampie, vuole riassumere e riferirsi a tutte quelle problematiche incontrate nello sviluppo dei sistemi di rappresentazione, archiviazione, interrogazione e reperimento delle informazioni di banche dati ad ogni livello di precisione e specificità.

Esso avrebbe indicato anche tutte quelle strutture specificatamente dedicate e necessarie alla gestione di corpora multilingue, e dunque l'identificazione della lingua di redazione,[21] la decodifica dei caratteri,[22] strutture di indicizzazione di corpora testuali multilinguistici, tecnologie di *Cross-Language* (o *Cross-Lingual*) *Information Retrival*,[23] e tutte quelle strutture da implementare per lo sviluppo della fase successiva al reperimento dei documenti, e cioè la

---

Il funzionamento di un sistema di ricerca delle informazioni si articola in tre compiti fondamentali:

1. la creazione e il mantenimento di indici;
2. la ricerca;
3. la presentazione dei risultati.

Anche se, per chi lo usa, l'importanza maggiore va all'efficacia della presentazione, il ruolo predominante è quello della costruzione degli indici.

Un'operazione di ricerca prende di solito le mosse da un'interrogazione da parte di un utente; il risultato ideale è l'insieme dei documenti che la soddisfano.

Il significato di "soddisfare un'interrogazione" dipende dalla forma di questa, dalle intenzioni di chi interroga e da altre caratteristiche del sistema. Nel caso più semplice, può significare "contenere una o più delle parole che formano l'interrogazione".

In casi più complessi, si può giudicare se un documento soddisfa un'interrogazione solo con l'uso di tecniche di trattamento del linguaggio, tecniche statistiche, di psicologia del lavoro e altro.

Infine, il risultato deve essere presentato all'utente in modo utile, sia per una comprensione immediata sia per poter formulare di nuovo l'interrogazione rapidamente e seguendo una strategia di ricerca.

Un'interrogazione può essere anche implicita: è il caso dei cosiddetti "recommender systems", il cui compito è analizzare gli interessi dell'utente (ad esempio, registrando sessioni di ricerca precedenti e documenti già consultati), estrarre dei termini chiave con i quali avviare una ricerca autonomamente - tecnicamente in "batch mode" - e presentare periodicamente i risultati.

Ogni operazione di ricerca fa cardine su un indice, che può essere anche costruito automaticamente.

Infatti, se la collezione di documenti è in forma elettronica, degli indicizzatori automatici possono sfruttare in vario modo le parole presenti nei testi e la loro rilevanza - data, ad esempio, dal numero di occorrenze o dalla posizione nel testo - per costruire degli indici. I cosiddetti motori di ricerca disponibili in rete sono costruiti a partire da indicizzatori automatici.

[21] Sia della lingua di redazione dei documenti archiviati, sia del linguaggio usato dall'utente esecutore dell'interrogazione e quindi destinatario dei dati estratti dal web.

[22] Il problema di UNICODE

[23] *C.L.I.R.*

visualizzazione, e dunque la presentazione, la sommarizzazione e la traduzione automatica dei risultati di una interrogazione.[24]

Dunque lo studio e lo sviluppo di strumenti e tecnologie per il *MultiLanguage Information Access* riguarda un'area di studio multidisciplinare dove sarebbero confluite metodologie e strumentazioni sviluppate nel campo del *Natural Language Processing* (*Multiple Language Recognition, Manipulation* e *Display*) e dell'*Information Retrival* (*Multilanguage* o *Cross Language Search* e *Cross Language Retrival*).

[24] La precisa definizione terminologica delle accezioni attribuite al termine ***M.L.I.A.*** è ardua poiché essa riguarda la definizione di una nuova area multidisciplinare caratterizzata da una terminologia non ancora del tutto stabile e definita. A volte il termine ***M.L.I.A.*** è affiancato ad altri gruppi di applicazioni, come ad esempio il *MultiLingual Information Retrival*, invece riferito a quel gruppo di sistemi sviluppati per il recupero delle informazioni monolingue relative eccetto l'Inglese, come il sistema di *Text REtrival Conferences*, T.*R.E.C.*, implementata per operazioni di *I.R.* sulla lingua ***spagnola.***
Altro termine introdotto dall'agenzia *D.A.R.P.A.., US Defence Advanced Research Projects Agency*, è *Translingual Information Retrival*, usato per indicare una serie di tecnologie, tra le quali il *C.L.I.R.*, dedicate al reperimento, la visualizzazione e l'amministrazione di *corpora* di documenti multilinguistici.

2.2 Natural Language Processing

L'elaborazione automatica del linguaggio naturale, *Natural Language Processing* o *N.L.P.*, consiste nello sviluppo di modelli ed algoritmi per la simulazione del processo linguistico umano.

Tale definizione lascia intravedere un campo sconfinato di applicazioni possibili e di multidisciplinarietà degli approcci e delle tecnologie implementabili.

Tuttavia, le principali applicazioni sviluppate per la realizzazione di alcuni dei processi classificabili come appartenenti alla linguistica umana, e quindi raggruppabili sotto tale area, potrebbero essere ridotte a due principali tipologie atte a:

- *permettere la comunicazione uomo-macchina* e *migliorarne l'interazione*: settore nel quale possono essere raggruppate tutti i sistemi di *speech recognition-understanding* ed *interfacce per il linguaggio naturale*.

- *migliorare la comunicazione uomo-uomo*: come la costruzione di strutture e sistemi per la *classificazione automatica di corpora testuali*, sistemi di *rappresentazione* e di *sommarizzazione* delle informazioni, sistemi di *traduzione automatica*, sistemi di *Multilanguage Information Access*, sistemi di *Information* e *Cross Language Information Retrival*.

Principale interesse della ricerca e` l'individuazione e l'implementazione di tecniche per lo sviluppo automatico di sistemi per il *N.L.P.* che abbiano un grado di accuratezza paragonabile a quella di sistemi prodotti manualmente, e tali tecniche sono usualmente basate sull'analisi automatica di corpora di testi di grandi dimensioni.

Tra gli argomenti a cui la ricerca ha dedicato maggior interesse possono essere annoverati i seguenti:

- *Parsing*: analisi sintattica automatica della frase; tale fase risulta preliminare per qualsiasi applicazione basata sulla comprensione automatica,

- *Part of Speech Tagging*: tale applicazione consiste nell'assegnare a ciascuna parola di un testo la corretta categoria sintattica, risolvendo le possibili ambiguità sulla base dell'analisi sintattica e contestuale,

- *Information Extraction*: assegnato un testo su un dominio specifico, si vuole sintetizzare l'informazione secondo uno schema preassegnato, ricavandone una sommarizzazione dei contenuti,

- *Machine Translation*: traduzione automatica di testi, per adesso implementata con discreti risultati su particolari tipi di testi, quali istruzioni tecniche, annunci, pagine web, ecc…, tutti elementi di corpora appartenenti a domini specifici o comunque dal vocabolario molto settoriale, caratteristica fondamentale per limitare le scelte del processo di disambiguazione dei termini e del loro contenuto semantico.

2.3 Sistemi di Information Retrival

I sistemi di *Information Retrival*, o *I.R.S.*, sono quelle tecnologie sviluppate all'interno del settore informatico mirate alla risoluzione ed al trattamento dei problemi relativi alla memorizzazione, rappresentazione e reperimento di *documenti*.

Questa definizione, tuttavia, potrebbe risultare ingannevole, poiché sembrerebbe limitare i target possibilmente oggetto di operazioni di *I.R.* al solo trattamento di *documenti testuali*, mentre l'applicazione di sistemi di *Information Retrival* non è così ristretta, ed è semplicemente una concausa di motivi pratici e storici ad aver comportato che la maggior parte dei sistemi e delle tecniche siano di fatto relativi a corpora testuali.

Storicamente, le applicazioni di *I.R.* nacquero per far fronte ai problemi sollevati dalla gestione di letterature specialistiche, più o meno vaste, e per soddisfare le richieste effettuate dagli utenti interessati a reperire informazioni *rilevanti* rispetto alle loro esigenze.

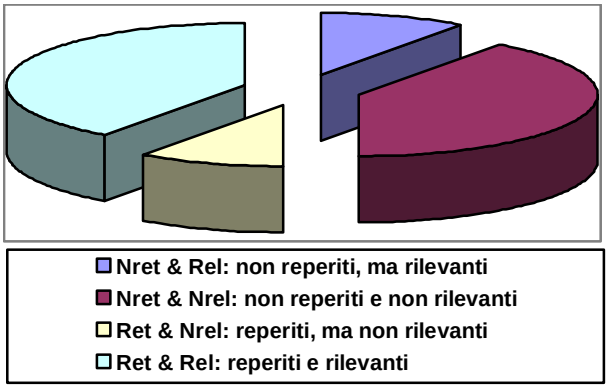
In un *I.R.S.* è importante considerare due qualità derivate dalle tecnologie implementate:

- *l'efficienza*, ovvero come il sistema si comporta in termini di tempi di risposta, uso della memoria, ecc…,

- *l'efficacia* (*effectiveness*), ovvero quanto il sistema è in grado di soddisfare l'utente fornendogli le (sole) informazioni rilevanti e semplificando il più possibile la sua attività di indagine conoscitiva.

Il problema di valutare l'efficacia di un *I.R.S.* non è di facile soluzione, in quanto include diversi aspetti soggettivi.[25]

Si supponga, ad esempio, di poter definire per ogni documento in una data collezione se esso sia rilevante o meno in riferimento ad una data interrogazione,[26] e di schematizzarne il risultato sia del *retrival* sia della pertinenza alle interrogazioni: l'insieme dei documenti appare suddiviso in quattro sottoinsiemi, dove la divisione (e quindi la classificazione di un documento) dipende dal fatto che lo stesso sia stato reperito o meno dal sistema di *I.R.S.* e dalla sua rilevanza nei confronti della *query*,[27] come schematizzato nel grafico.



I documenti *Ret & NRel* chiamati anche *Falsi Positivi*, costituiscono il cosiddetto *rumore*, che ogni *I.R.S.* dovrebbe cercare di ridurre al minimo: tali documenti sono anche detti *false drops* o *false alarms* o ancora *false hits*.

I documenti *NRet & Rel*, viceversa, sono quelli per cui il sistema rimane silenzioso ed anch'essi, per i quali si usa anche il termine *false dismissals*, dovrebbero essere portati al minimo: in conclusione un *I.R.S.* efficace dovrebbe tendere a non produrre *Falsi Positivi* e *false dismissals*.[28]

Le due misure più comuni per quantificare l'efficacia di un *I.R.S.* sono allora *Recall* e *Precision*:

---

[25] Non per ultimo il livello di specializzazione che gli utenti possono avere nel dominio di ricerca effettuata: ad esempio, due utenti con diversi livelli di conoscenza, a priori, formulando la stessa richiesta, potrebbero fornire diverse valutazioni sull'insieme di documenti reperiti dal sistema.

[26] L'interrogazione si definisce *Query*.
Per la valutazione delle qualità di un I.R.S. esistono collezioni test di documenti e richieste per cui sono disponibili valutazioni di rilevanza, ed è possibile quindi valutare il sistema di I.R. a cui il *corpus* di *query-documenti* è applicato.

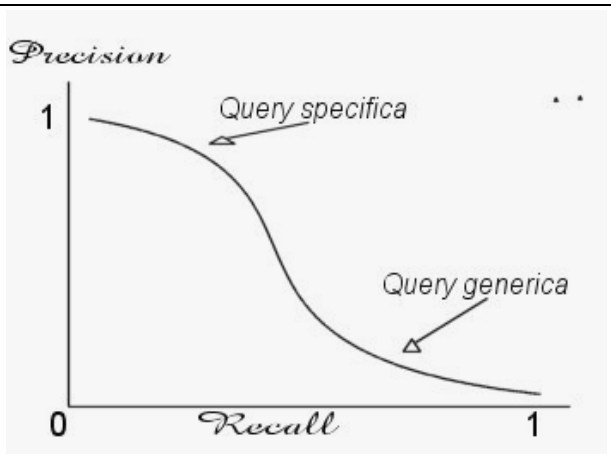
[27] Con il termine *query* si intende una stringa digitata per un interrogazione.

[28] E cioè avere *no false drops* e *no false dismissals*.

$$\text{Recall:} \quad \frac{R = \#\,Ret\ \&\ Rel}{\#RelTot}$$

$$\text{Precision:} \quad \frac{P = \#\,Ret\ \&\ Rel}{\#RetTot}$$

In genere, aumentando il *Recall* la *Precision* diminuisce e viceversa, seguendo un andamento graficamente rappresentato di seguito:



I.R.S. ideale :

(*R*,*P*) = (1,1)

Si noti che, mentre la *precision* è calcolabile a partire dal risultato (*#RetTot*), così non è per il *recall*, che richiede di conoscere quanti sono i documenti rilevanti in tutta la collezione (*#RelTot*).

Supponendo di conoscere *#RelTot*, l'efficacia di un *I.R.S.* viene normalmente valutata misurando la *precision* a diversi livelli di *recall*, ovvero è necessario calcolare quanti documenti si debbano reperire (*#RetTot*) affinché il risultato contenga una frazione pari a *R* nei documenti rilevanti presenti nel corpus esaminato (*#RelTot*).

$$P = R\ x\ \frac{\#RelTot}{\#RelTot}$$

È ovvio che la valutazione completa deve essere eseguita su un certo numero di *queries*, e che essse debbano essere significative, avendo quindi i presupposti necessari per cui al termine dei test sul sistema di *I.R.* siano disponibili giudizi rilevanti.

Ogni *I.R.S.* si basa quindi su una *tecnica di recupero* delle informazioni, ovvero su un meccanismo che permette di confrontare la richiesta, espressa in

uno specifico *linguaggio*, con le *rappresentazioni* dei documenti, coincidenti con i documenti stessi (*rappresentazione diretta*) o date da *surrogati* (*rappresentazione indiretta*) più o meno efficiente, e tutti gli sforzi compiuti sia nello studio dei problemi direttamente implicati, sia riguardo a tutto quell'insieme di architetture indirettamente coinvolte nei sistemi di *I.R.*, sono dunque indirizzati verso l'incremento della sua efficacia.

In base alle soluzioni adottate sia nelle strutture relativa alla ricerca ed all'estrazione dei documenti in base ad una *query*, sia nelle architetture tanto degli *I.R.S.* quanto delle base di dati contenenti i *corpora* di documenti, è possibile individuare diversi aspetti generali in base ai quali si comportano i sistemi di *I.R.*, e questi aspetti possono essere ricondotti a:

3. *Modello concettuale*: Il *modello concettuale* definisce la *filosofia di fondo* di un *I.R.S.*, ovvero attorno a quali principi generali si è sviluppato ed opera il sistema.
   L'uso di un *modello concettuale* influenza e determina principalmente *il linguaggio di interrogazione*, *la rappresentazione dei documenti*, *la struttura dei file*, *i criteri di reperimento dei documenti*.
   I modelli concettuali sono a loro volta divisibili in alcuni sottomodelli:

   - *Booleano*: i documenti sono rappresentati da insiemi di termini chiave (*keywords*), estratti manualmente e/o automaticamente dal testo, e le richieste sono keyword connesse da operatori logici; sono i sistemi più diffusi, a motivo della loro semplicità ed efficienza, ma possono presentare problemi in termini di efficacia,

   - *Booleano esteso*: caratterizzato per la presenza di pesi (*weight*) associati ai termini di un documento, che ne riflettono l'importanza relativa all'interno del documento; permette inoltre di adottare criteri di ordinamento (*ranking*) dei documenti reperiti.
     La pesatura dei termini in un sistema di *IR*, potrebbe procedere come quanto esemplificato di seguito: la *query controllo delle piogge acide nella Foresta Nera* potrebbe venire riformulata come *Controllo AND Pioggia AND Acido AND Foresta Nera* e porterebbe a reperire i documenti che contengono *tutti* i termini della *query* indipendentemente da:

- l'importanza relativa dei termini stessi nella *query* (utile se, ad esempio, la *query* avesse contenuto relazioni di tipo *OR*)
- l'importanza dei termini nei documenti (utile in fase di indexing e di ranking)

In molti casi non è ben chiaro cosa specificare nella *query*, e questo può portare o a produrre *query* troppo generiche (*precision bassa*) o troppo specifiche (*recall basso*) come mostrato di seguito seguendo il precedente esempio:

- *specifica*: *algoritmi di allocazione dati su reti di calcolatori* che non genera alcun risultato (no matches),
- *generica*: *algoritmi per reti di calcolatori* che genera molto materiale retrived (400 matches)con molto rumore relativo rispetto alle intenzioni di ricerca dell'utente.

Esistono molti criteri di pesatura dei termini; il più noto è *tf.idf*, ovvero *term frequency and inverse document frequency*.
Altro modello da annoverare è il così detto *M.M.M.*, ossia *Mixed Min and Max*.
M.M.M., nonostante non sia il migliore tra i modelli *Booleani estesi* in termini di aumento medio di precision, ha il vantaggio di essere computazionalmente quello meno oneroso.
Ad esempio il modello detto di *P-norm* ha il pregio di considerare tutti i termini di un documento, e non solo quelli con peso minimo e massimo.
I miglioramenti in *precision* vanno dal *79%* al *210%*; il modello può risultare *computazionalmente costoso* a causa delle operazioni con esponenziali richieste.

- *Vector space*: i documenti vengono visti come *punti* in uno spazio le cui coordinate sono tutte le *keyword* gestite dal sistema, e i pesi determinano il valore di tali coordinate.
  Nel modello *Vector Space* sia il documento che la *query* sono vettori (non si fa pertanto uso di operatori *Booleani*) i cui componenti sono i pesi dei termini usati per l'indicizzazione.

- *Probabilistico*: assegna pesi ai termini considerando la loro probabilità di essere presenti in documenti rilevanti ad una data *query*.[29]

- *Clustering*: i documenti sono organizzati, sulla base del loro contenuto, in *cluster* omogenei secondo una certa metrica. Il sistema ne può guadagnare sia in termini di efficienza, confrontando una *query* solo con i *rappresentativi* dei *cluster*, sia in termini di efficacia, considerando che l'associazione in cluster fornisce informazioni sulla rilevanza dei documenti.
  Un *cluster* è un gruppo omogeneo di documenti che sono tra loro più fortemente associati tra loro di quanto lo siano con documenti in altri gruppi; l'efficacia dei cluster si basa sulla cosiddetta *clustering hypothesis*: *documenti fortemente associati tendono ad essere rilevanti o meno per una stessa query*.
  In una ricerca basata su *cluster*, il confronto avviene in due fasi, considerando prima i soli documenti rappresentativi dei *cluster* (*centroidi*) e quindi i documenti nei soli *cluster* selezionati.
  Il *clustering* può essere inoltre di tipo *gerarchico* (organizzazione ad albero dei *cluster*) o a *singolo livello* (con *overlap* tra *cluster* o meno).

- *String search*: la tecnica di recupero opera semplicemente con algoritmi di *pattern-matching* su *query* e documenti.

2. *Strutture dei files*: il sistema può limitarsi a gestire semplici files sequenziali nel caso di ricerche di stringhe, oppure può fare uso di diverse tipologie organizzative tra cui le più importanti possono essere riassunte nelle due seguenti:

   - *Inverted File*: l'inverted file è un indice ordinato e per ogni keyword riporta la lista dei documenti che la contengono (eventualmente anche la/le posizione/i nel testo),

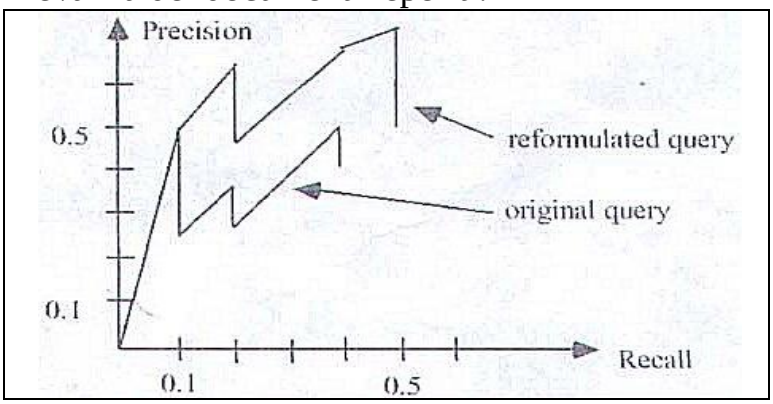
[29] Teoricamente interessante, ma necessita la conoscenza per lo meno di minimo livello, a priori, dei documenti rilevanti

- *Signature File*: un *signature file* , anche *S.F.*, memorizza, sotto forma di stringhe binarie, delle *astrazioni* (o *signature*) dei documenti, che vengono confrontate con una corrispondente signature della *query*.
  La tecnica di recupero basata su *S.F.* è di tipo *booleano* (documenti che contengono i termini nella *query*), ma non ha un grado di precisione elevato.

3. *Operazioni sulle query* : Esistono *I.R.S.* che accettano anche input in linguaggio naturale; in questo caso vi è essenziale un'attività di *parsing* che estrae i termini rilevanti e li connette attraverso opportuni operatori (*Booleani*).
   Il tipo di operazione sulla *query* può essere distinto in:

   - *Semplice*: il sistema di *I.R.* manipola la stringa di *query* (tramite un *parser*) ricostruendola in un'*espressione booleana* le cui relazioni dipendono dalle opzioni, se fornite, di ricerca all'interno del database;

   - *Feedback*: un *I.R.S.* può anche fornire meccanismi di *query feedback*, attraverso i quali i documenti reperiti e giudicati rilevanti dall'utente vengono usati per riformulare la *query*.
     L'idea sulla quale si basano i meccanismi di *query feedback*, detti anche di *relevance feedback*, consiste nel modificare la *query* usando l'informazione acquisita dall'utente (primi documenti estratti dalla *query* originale), sulla rilevanza dei documenti reperiti.



     La parte di documenti estratta tramite la prima *query*, in pratica, viene utilizzata come una seconda

query per una ulteriore interrogazione verso i corpora.
Applicazioni di questa tecnica (o con tecnologie simili) portano sperimentalmente a miglioramenti nella *precision*, per determinati livelli fissati di *recall*, che possono variare dal 40% al 60%, come raffigurato nel grafico.

4. *Operazioni sui termini*: la prima operazione che un *I.R.S.* deve eseguire consiste nell'attività di individuazione di *keywords* (automatica e/o manuale) denominata *indexing*.
   Al fine di migliorare l'efficacia del sistema sono stati implementate diverse tecnologie, tra le quali è doveroso citare:

   - *Stemming*: i termini vengono ricondotti ad una radice comune, ad esempio rimuovendo prefissi e suffissi; lo stemming migliora il *recall* e riduce la dimensione degli indici.
     Un particolare processo di *Stemming* può generare una *indicizzazione linguistica* dei documenti, il cui processo è chiamato *Linguistically Motivated Indexing*, o *L.M.I.*.[30]

   - *Ontologie* e *Thesaurus*: si fa uso di un vocabolario di sinonimi e di termini correlati.
     Il *thesaurus* è lo strumento terminologico principe per l'organizzazione della conoscenza; utilizzato soprattutto per la classificazione e consultazione dei grandi archivi bibliografici in ogni settore scientifico letterario o artistico, è la chiave per l'accesso razionale, ad esempio, alle principali biblioteche ed ai relativi sistemi documentari.

[30] La nozione di *L.M.I.* si basa sull'interpretazione delle voci di indice come veicoli di informazioni rappresentate da chiavi.
Una volta accertato che le stesse *chiavi* siano condivise tra la *query* ed il documento si può presumere che almeno una parte delle informazioni contenute nel documento siano rilevanti per l'interrogazione.
La *L.M.I.* implica, per lo meno, che vi siano state a priori delle tecnologie in grado di analizzare i contenuti semantici dei testi, generare un indice di chiavi, e connettere queste ultime a tutti quei testi riconducibili alle stesse chiavi.
L'indicizzazione dei documenti avrebbe potuto avvenire prima e indipendentemente dall'interrogazione verso il database, oppure che essa sia stata catalizzata proprio dall'interrogazione verso un determinato corpus di informazioni.
L'applicazione di un *L.M.I.* riguarda, naturalmente, anche il testo componente la *query* dell'utente.

Ognuno di questi grandi sistemi è dotato di un proprio *thesaurus*, le cui dimensioni contenuto e struttura sono stati sviluppati in funzione del rispettivo patrimonio e dei propri criteri organizzativi.
Il *thesaurus* è dunque un *soggettario standardizzato* all'interno del quale un concetto viene sempre ricondotto ad un unico termine, indipendentemente dalle varianti linguistiche che diversi autori, ad esempio, potrebbero utilizzare per rappresentarlo.
Esso, in fase di indicizzazione, è lo strumento utilizzato dai catalogatori, umani, automatici o semiautomatici, per attribuire i soggetti ai records bibliografici al fine di rappresentarne il contenuto in modo univoco.
In fase di ricerca è uno degli strumenti che consentono il livello piu' alto di controllo dei termini che andranno a comporre la strategia di ricerca.
Il *thesaurus* è quindi una struttura che organizza le complessità delle terminologie di un linguaggio e provvede alla rappresentazione delle relazioni concettuali tramite un'*ontologia*.
Un'*ontologia* può essere definita come *un sistema di concetti rilevanti per una modellizzazione di un dato dominio di conoscenza*, e dunque consiste in una collezione di concetti organizzata secondo diversi tipi di relazioni, ed secondo una classificazione gerarchica dei termini.
Le *ontologie*, se adeguatamente sviluppate, estese e dettagliate, permettono sia applicazioni soddisfacenti per operazioni di *IR*, sia per applicativi di *Machine Translation*, ma la caratteristica che le distingue è la qualità di essere *language-neutral*, poiché esse, se ben realizzate, non contengono parole, ma puri concetti. [31]

[31] Fra le ontologie implementate è doveroso citare:
CYC: il progetto è stato intrapreso per la realizzazione di un sostrato rappresentante una conoscenza di "significati generali" come supporto per applicativi di intelligenza artificiale. *CYC* è sfruttabile anche come ontologia standard (vocabolario e schema strutturale di concetti) applicato all'*IR*.
WordNet: sviluppato presso l'*Università di Princeton* il progetto consiste nella realizzazione di una tassonomia di sensi monolingue (Inglese) ispirato alle attuali teorie psicolinguistiche circa il

Un *thesaurus* prevede quindi collegamenti semantici espliciti tra termini.
Un esempio di *thesaurus* embrionale e rudimentale può essere fornito dalla guida per la consultazione delle *Pagine Gialle*: i termini usati per l'identificazione delle categorie merceologiche elencate nelle pagine gialle consiste in un indice nel quale sono elencati tutti termini ufficiali di questo linguaggio che, per le sue limitazioni e caratteristiche apposte, è detto linguaggio limitato.
Le *Pagine Gialle*, dunque possono essere considerate come un indice in cui sono elencati tutti i termini ufficiali del suddetto linguaggio controllato.
I *thesauri* possono essere *monolingua*, e ve n'è un numero discreto, e *multilingua*, dei quali solo adesso se ne conta un certo incremento, soprattutto in un *contesto europeo*.

- *Stoplist*: lista di parole comuni che non hanno alcun valore selettivo e vengono pertanto eliminate in fase di *indexing*;[32]

- *Weighting*: le *keywords* vengono pesate numericamente facendo uso di informazioni sulla distribuzione statistica dei termini;

5. *Operazioni sui documenti*: un documento può ovviamente contenere parti strutturate, quali autore, data, ecc..; considerando ciò, un *I.R.S.* può eseguire le seguenti operazioni su ognuna delle strutture presenti:

   - *parsing*: individua i *campi* costituenti ed i relativi valori

---

funzionamento della memoria lessicale umana. Aggettivi, sostantivi, predicati ed avverbi sono stati organizzati in set di sinonimi (detti synset) ognuno dei quali rappresenta un determinato concetto. I diversi synset sono organizzati fra loro da diverse classi di relazioni.
EuroWordNet: basato su *WordNet*, il progetto consiste in un database multilingue basato su un'architettura detta di TopOntology che ha generato un sistema di relazioni semantiche di base fra synset di 8 lingue europee. All'interno di ciascun WordNet sono state mantenute le specifiche del singolo linguaggio.

[32] Per lo sviluppo dell'argomento delle Stoplist vedi di seguito

- *field masking*: "occultamento" di campi (*proiezione*)
- *ranking*: ordinamento sulla base della rilevanza[33]
- *clustering*: classificazione in *cluster*.[34]

Queste le principali funzioni e strutture di cui deve essere dotato un *I.R.S.* efficiente; di seguito saranno esaminate nel dettaglio alcune caratteristiche ed applicazioni avanzate e specializzate.

2.4 Multilanguage Text Processing

2.4.1 Comprensione del linguaggio

La prima funzione che un sistema di *Multilanguage Text Processing*, *M.T.P.*, deve aver implementata consiste nella capacità di comprensione del linguaggio del documento esaminato, e tali strutture sono state sviluppate seguendo diversi approcci comunque in genere capaci di riconoscere all'interno delle stringhe particolari sequenze di caratteri significative di determinati linguaggi, così come quei sistemi che cercano all'interno del testo esaminato la presenza delle *stopwords* specifiche di ciascun linguaggio che sono, per natura sintattica, le parole più facili e ricorrenti da incontrare all'interno di un testo.

La codifica di un linguaggio, comunque, dipende da una precedente capacità del sistema analizzante, di riconoscimento e codifica del set di caratteri corrispondente all'alfabeto nella sua rappresentazione binaria.

Questo schema di corrispondenza fra alfabeto scritto e la sua rappresentazione varia strutturalmente per i diversi linguaggi: ad esempio i caratteri cinesi, ideografici, richiedono un sistema di codifica *double-byte*, mentre altre lingue si basano su soluzioni ad *un byte singolo*.[35]

Nell'analisi testuale, se il processo di codifica è necessario, un sistema si appoggia allo standard *UNICODE* per convertire le singole codifiche locali dei caratteri del linguaggio in un formato standard costituito da una rappresentazione univoca.

---

[33] La rilevanza non può essere considerata certa ed oggettiva: essa è semplicemente presunta.

[34] Vedi *Modello concettuale a Clustering*

[35] In attesa di un solo schema di codifica per la mappatura di tutte le lingue mondiali, il consorzio UNICODE ha prodotto lo standard UNICODE per provvedere ad un sistema di codifica dei caratteri designato al supporto dell'interscambio, dell'analisi e della visualizzazione di testi scritti in lingue diverse, moderne e antiche (per la visualizzazione di testi classici e storici). Nella sua versione più recente lo standard UNICODE è costituito da 38.887 codifiche di carattere, riuscendo così a coprire le principali lingue scritte americane, europee, del Medio Oriente, africane, indiane asiatiche e del Pacifico.

2.4.2 Tokenizzazione

Una volta eseguita, se necessaria, la codifica dei caratteri, e riconosciuta la lingua in cui il testo è redatto, un sistema di *M.T.P.* deve essere in grado di identificare le singole parole all'interno delle stringhe di testo.

Per molte lingue questo è facilmente implementabile poiché esse usano, in linea di massima, il carattere "*_blank*", spazio, come separatore, ma altrettante sono caratterizzate da maggiori difficoltà perché le parole sono concatenate fra loro per la composizione di termini particolari, o addirittura non presentano il carattere di spazio come separatore (ad esempio il *Giapponese*, il *Cinese*, ecc…), e dunque il processo di *tokenizzazione* deve essere capace di riconoscere i limiti di ciascuna parola: in questo caso sara necessario ricorrere ad un sistema di dizionari.

2.4.3 Eliminazione delle Stopwords

Nell'ordine di ridurre al minimo la quantità di testo indicizzato vengono cercate ed eliminate le parole che hanno un valore minimo o nullo nella rappresentazione del testo originale (*stopwords*).

Il processo di eliminazione delle *stopwords* è estremamente proficuo, e riesce ad eliminare dai testi a cui viene sottoposto dal 30% al 50% delle parole: esso si basa su un archivio che raccoglie tutte le possibili *stopwords* (detto *stoplist*) come articoli, preposizioni semplici e composte, ecc…, e gli elementi di ogni *stoplist* possono variare e dipendere anche dal dominio dei *corpora* di testi analizzati.[36]

2.4.4 Normalizzazione e Stemming

L'ultimo stadio del processo di indicizzazione dei testi per *I.R.S.* consiste nella *normalizzazione* delle parole del testo rimanenti dopo la *tokenizzazione* e la rimozione delle *stopwords*.

La forma di normalizzazione più comune consiste nella rimozione dei suffissi e delle flessioni, riportando così le varie parole alla loro forma di radice (detta *stem*, da cui prende nome il processo, definito, di *stemming*).

Nel caso più semplice vi è un algoritmo che rimuove i suffissi standard, e la sua implementazione dipende dal linguaggio per il quale è stato costruito il processo di *stemming*, ad esempio per l'*Inglese* il processo individuerà ed eliminerà con un processo iterativo i suffissi e le flessioni del plurale "*s, es, ecc...*", fino a quando il termine non sia portato alla sua unità minima di radice.[37]

La maggior parte degli algoritmi di *stemming* sono stati prodotti per la normalizzazione su documenti di lingua inglese, e quindi la loro applicazione ad un contesto di lingue europee ha provocato una serie di problemi relativi alle flessioni morfologiche più ricche di quanto abbia la lingua inglese.

La realizzazione delle strutture necessarie ad una adeguata indicizzazione di testi multilingue è naturalmente applicabile anche ai testi delle *queries*, ed è quindi la base per un il reperimento di informazioni multilingue, detta *Cross-Language Information Retrival*, o *C.L.I.R.*.

2.5 Cross-Language Information Retrival

---

[36] ad esempio nel caso specifico, un corpus di testi egittologici, dedicato al tema delle piste carovaniere del deserto occidentale potrebbe annoverare all'interno delle proprie stoplist le parole egittologia, Egitto, egiziano, occidentale, ecc...

[37] L'algoritmo di *stemming* non è sempre così "*banale*": solo in alcuni casi è possibile ridurre un termine alla propria *stem* con un'operazione cruda (ad esempio *organico – sic = organi*), e quindi deve procedere a soluzioni alternative, come utilizzare una analisi morfologica del testo, e tramite essa ridurre le parole alla loro radice lessicale, come se queste apparissero in un dizionario standard.

In generale un sistema adeguato di *C.L.I.R.* deve contenere delle tecnologie capaci di relazionare correttamente *queries* e documenti a prescindere dal linguaggio di redazione di entrambe, e quindi disporre i documenti in ordine di *rilevanza*.[38]

A differenza di un sistema di *retrival monolingua*, in un *C.L.I.R.s.* è necessario sopperire ad un necessario processo di *words matching* e *weighting* da applicare a più linguaggi.

Questo implica tra l'altro un precedente sviluppo di una risorsa lessicale capace di tradurre dal linguaggio delle *queries* a quello/i del documento e viceversa, e che sia capace di risolvere il problema delle ambiguità, già pesantissimo all'interno di sistemi di *retrival monolingue*, ed enormemente amplificato in un contesto *plurilinguistico*.

Sono stati sviluppati principalmente tre tipi di approcci per la realizzazione di sistemi di *C.L.I.R.* tramite altrettante tecniche:

1. *Machine Translation*: sebbene il pregio di un sistema di *Machine Translation* sia la produzione di testi comprensibili ed affidabili da un linguaggio *sorgente* ad un linguaggio *target* (premessa ed ammessa la totale affidabilità del sistema di *M.T.*), è una soluzione che non è vista come una risposta reale al problema di sopperire al *matching* e *retrival* di documenti *multilingue*.
   Tramite un sistema di *M.T.* sarebbe possibile sopperire alla riduzione di una *query* e dei documenti relativi ad uno stesso linguaggio per poi ponderare se i diversi documenti fossero rilevanti verso le informazioni espresse nelle *queries*: ovviamente il processo di *M.T.* avrebbe riguardato la traduzione del testo di queste ultime.
   Tuttavia la traduzione accurata dei testi delle *queries* può essere vista sia come impossibile da realizzarsi, ma anche come non necessaria, visto che nel caso specifico (produzione di *queries multilingue*) non c'è la necessita della produzione in linguaggio target dei termini della *query*, e molto spesso la traduzione da una *query* sorgente a *queries* multiple plurilingue può comportare un miglioramento nelle performance del sistema di *C.L.I.R.*.

2. *Knowledge-based*: la tecnica è sviluppata tramite l'applicazione di *thesauri*, *ontologie* e *dizionari bi-* o *multilingue*.

2.1. Thesauri ed Ontologie

[38] Anche in questo caso il sistema deve essere capace di distinguere quali siano le parti del *corpus* (i documenti) più importanti da visualizzare con priorità e quali siano le parti del testo di una *query* (le parole) più rilevanti da rendere più ponderanti nel sistema di *retrival*.

L'applicazione dei *thesauri* ad un sistema di *C.L.I.R.* è stata storicamente il primo tipo di soluzione adottata.
Riprendendo il concetto di *thesaurus* esaminato precedentemente, il suo utilizzo all'interno di tali sistemi potrebbe essere interpretato come lo sfruttamento di una *ontologia specializzata nell'organizzazione terminologica*, e quando esso è realizzato in un contesto multilinguistico prevede ovviamente l'organizzazione terminologica di tutti i termini di ciascuna lingua.
A sua volta *thesaurus multilingue* per l'indicizzazione e la ricerca di documenti può essere visto come un set di *thesauri monolingue*, tutti relazionati verso un sistema comune di concetti.
Tramite questo sistema, un utente è in grado di produrre un'interrogazione in una determinata lingua, ed ottenere i documenti contenenti i concetti corrispondenti negli altri linguaggi.
Attraverso questo sistema di approccio, possono essere assegnati per ogni documento i termini ad esso appropriati
Le sperimentazioni recenti, e la presenza in commercio di soluzioni basati proprio su questa tecnologia, hanno dimostrato che un sistema di *thesaurus* multilingue può fornire buoni risultati in un contesto di *C.L.I.R.*.
Gli obbiettivi della ricerca attuale, oltre che mirare allo sviluppo di *thesauri multilingue*, finora presenti in minor numero rispetto a quelli *monolingua*, sono finalizzati alla produzione di assegnazioni semiautomatiche dei concetti, precedentemente realizzata manualmente da uno staff di esperti, nel campo specifico.

2.2. Dizionari

Molti sistemi di *C.L.I.R.* utilizzano una serie di *dizionari bilingue* come *interfacce di traduzione*.
Generalmente i *dizionari bilingue*, predestinati ad un'utenza umana, se sottoposti ad un'operazione di *pre-processing*, possono essere resi utilizzabili da sistemi automatici, costituendo una serie di *Machine Readable Dictionaries*, *M.R.Ds.*.
Tuttavia le sperimentazioni di sistemi di *C.L.I.R.* basati su *M.R.Ds.* si sono dimostrate carenti, mancando dal *40%* al *60%* delle operazioni di *retrival* eseguite da *I.R.S.* monolingue: è stato calcolato che l'occorrenza '*out of vocabulary*' in sistemi di *M.R.Ds.* bilingue sia la causa del *23%* degli effettivi fallimenti di sessioni di *C.L.I.R.*.

3. *Corpus-based*: è un sistema di approccio realizzato tramite l'analisi statistica di ampi *corpora testuali* e l'estrazione automatica di informazioni necessarie alla costruzione di specifiche applicazioni di traduzione.
   La collezione di testi generalmente consiste in una serie di *corpus monolingue paralleli* per l'estrazione di termini multilingue equivalenti, e duna derivazione di questa tecnica, indipendente dai *corpora* analizzati.
   È la così detta tecnica di *Latent Semantic Indexing*, *L.S.I.*, capace di estrarre termini linguistici e di produrre rappresentazioni di documenti indipendenti dai *corpora paralleli*.
   Altro tipo di corpora utilizzabili per applicazioni di *C.L.I.R.* sono i *Comparable Corpora*, più semplici da reperire e da costruire dei *corpora paralleli*, e fra le applicazioni realizzate con tale tecnologia spicca il *corpus allineato Tedesco – Italiano* realizzato tramite l'analisi di storie della *Swiss news agency*, *S.D.A.*.
   La collezione di testi della *S.D.A.* è stata connotata manualmente di una serie di concetti detti *subject descriptors* realizzati ed attribuiti secondo lo stesso schema di classificazione.

2.6 Tecniche di Traduzione Automatica

2.6.1 introduzione storica

La *Traduzione Automatica* (*Machine Translation* o *M.T.*) consiste nella traduzione da un linguaggio naturale ad un altro tramite un sistema computerizzato, ed è stato uno dei problemi di maggiori difficoltà per circa 40 anni nello studio dell'intelligenza artificiale, specificatamente nel trattamento automatico del linguaggio naturale.

I primi tentativi di traduzione automatica fallirono a causa dell'interazione di fenomeni linguistici complessi per i quali sembrava impossibile raggiungere una traduzione efficiente.

I tentativi più recenti hanno invece raggiunto dei risultati notevoli, anche se è possibile affermare che la maggioranza dei sistemi di *M.T.* non producono una traduzione completamente "automatizzata", e il livello di attendibilità del prodotto è accettabile, ma necessita revisione post-traduzione.

Fra gli anni '50 e '60, sia in Europa che negli U. S. A., vennero compiuti grandi sforzi nel tentativo di automatizzare per lo meno alcune fasi del processo di traduzione: questi sperimentazioni erano eseguiti tramite semplici dizionari bilingue indicizzati, banche dati terminologiche, ed altri sistemi, ma malgrado il lavoro alacre alla *M.T.*, nel 1966 l'*A.L.P.A.C.*, *Automatic Language Processing Advisory Committee*, bocciò i risultati sino ad allora raggiunti perché di poca qualità ed estremamente costosi.

La bocciatura dell'*A.L.P.A.C.* costituì un taglio nella ricerca, soprattutto negli Stati Uniti, per la *M.T.*, alla quale vennero favorite altre aree ad essa però

sempre legate, come la *Linguistica Computazionale* e l'*Intelligenza Artificiale*, che successivamente fornirono delle fondamenta teoriche maggiori e delle tecniche migliori, che probabilmente furono anche una delle cause del fallimento dei primi applicativi di *M.T.*.

La ricerca specifica proseguì a livello accademico presso *l'Università* di *Austin*, *Texas*, *l'Università di Young* a *Brigham* e quella di *Georgetown*, per quanto riguarda gli U. S. A., e contemporaneamente venivano raggiunti risultati significanti in *Europa* ed ex *U.S.S.R.*.

Alla fine del 1960 anche il *Canada* iniziò una propria ricerca sulla *M.T.* necessaria a compensare la necessità di tradurre documenti ed informazioni, nel proprio caso specifico, in *Inglese* e *Francese* data la sua natura bilingue.

In *Europa* un passo importante nello sviluppo della *M.T.* fu rappresentato dal progetto *EUROTRA* della *Comunità Europea*, che si prefiggeva di realizzare dei sistemi di *M.T.* per coprire le lingue delle nazioni aderenti.

In *Asia* un fenomeno altrettanto ponderante fu la presa di coscienza del *Giappone*, della *Cina* e degli altri paesi, dell'importanza che avrebbe avuto sull'economia del paese la realizzazione di un sistema di *M.T.* per la traduzione di informazioni da fonti straniere.[39]

EUROTRA fallì nella realizzazione del sistema prefisso, ma da questo momento si affacciarono sul mercato software dedicati a talune coppie di lingue.

La ricerca *Giapponese*, nel frattempo, aveva prodotto una sovrabbondanza di prototipi di sistemi di *M.T.* commerciali, molti dei quali basati su tecnologia abbastanza stabile: la ricerca nipponica sulla *M.T.*, sebbene mai estensiva, è sempre cresciuta sia in qualità che in traguardi raggiunti.

Per quanto riguarda gli Stati Uniti, la ricerca ebbe un nuovo picco di sviluppo a partire dalla seconda metà del 1980, questo anche a causa della stimolante apertura di mercati stranieri, e dunque ripresero i finanziamenti alla ricerca di *M.T.* indipendentemente dagli studi di *Linguistica Computazionale*, prime fra cui l'*Università del New Mexico*, la *Carnegie Mellon University* e l'*Università del Maryland*.

All'interesse economico e militare, principale motore ed utente dei sistemi di *M.T.* sviluppati, oggigiorno assistiamo ad una vertiginosa espansione ad applicazioni della vita quotidiana e domestica (come le applicazioni che si basano su Comprensione e produzione del parlato), con il relativo sviluppo del mercato alla voce di compagnie private sul mercato mondiale.

2.7 sistemi di M.T.: i fattori linguistici

Le problematiche linguistiche di un sistema di *M.T.* possono essere raggruppate in 3 categorie:

1. *comprensione del linguaggio*,

[39] Prima di tutto europee e, poi, asiatiche.

2. *generazione del linguaggio*,

3. *Relazioni fra coppie di lingue* (linguaggio *sorgente* e linguaggio *target*).

Per quanto riguarda sia la comprensione del linguaggio sia la sua generazione, sono state proposte molte soluzioni ed i due problemi hanno avuto sviluppi ed approcci differenti: l'idea, ormai accantonata, era che per una corretta *M.T.* fosse necessaria una completa comprensione del *testo sorgente*.

Sia per ovviare alle difficoltà enormi della comprensione del *testo sorgente*, sia perché è stato dimostrato che il processo in realtà non è poi così strettamente indispensabile, in tempi più recenti la ricerca si è invece concentrata nel raggiungimento di una *M.T.* soddisfacente nella quale però l'analisi del linguaggio sorgente avesse un apporto minimo nella produzione del testo in linguaggio/i target, e quindi gli sforzi maggiori sono stati incentrati nel tentativo di risolvere problemi quali le *ambiguità sintattiche* e *lessicali*, *semantiche* e *contestuali*.

2.7.1 comprensione del linguaggio

Le ambiguità sintattiche e lessicali:

1. I *saw the man on the hill with the telescope*.
   Ho visto l'uomo sulla collina con il telescopio

In questo esempio non c'è alcun contesto linguistico che ci permetta di comprendere se il telescopio appartenga alla persona che parla o all'uomo a cui si riferisce (c'è anche una terza interpretazione: il colle sul quale si trova il telescopio) in questo caso, dato che l'ambiguità è al livello pragmatico, non c'è necessità di disambiguare e, trasferendo l'ambiguità al linguaggio target (laddove la sintassi della rispettiva lingua lo consente), essa sarà interpretata dal lettore.

2. *Inglese*: *book* **à** *Spagnolo*: *libro, reservar*.

Il sostantivo Inglese *book*, al momento della sua traduzione in Spagnolo, può essere tradotto come sostantivo se nella lingua sorgente compare dopo l'articolo, oppure col verbo *reservar*.

In questo caso abbiamo quindi un contesto grammaticale che può fornire le informazioni adeguate alla disambiguazione.

Le ambiguità semantiche:

Fra le ambiguità semantiche possono essere considerate quelle costituite dalla *Omografia* (alla stesa forma grafica corrispondono significati diversi), ad esempio:

3. *Inglese*: *ball* **à** *Spagnolo*: *pelota, baile*

dove il primo sostantivo indica un oggetto sferico, mentre il secondo si riferisce ad una danza, e la *Polisemia*, come il verbo inglese *to kill*, che contiene sottili differenze di significato in diversi contesti, ad esempio:

4. *Inglese*: *kill a man* **à** *Spagnolo*: *matar*, e

5. *Inglese*: *kill a process* **à** *Spagnolo*: *acabar*.

per la scelta della parola giusta nella lingua target si rende necessario allargare il contesto in cui essa compare, nel caso specifico, saranno gli argomenti del verbo a fornire le indicazioni.

Molto spesso le ambiguità semantiche sono così complesse che è impossibile, senza una profonda comprensione del testo (cioè senza ricorrere ad un contesto ampio che può corrispondere al limite ad un intero documento), riuscire a risolverle, come nell'esempio:

6. *Inglese*: *The computer outputs the result; it is fast*
   à *Spagnolo*: *La computadora imprime el resultado; es ràpida*.

Il contesto peculiare (*manuale informatico*), riduce alquanto il dominio del lessico utilizzato e quindi di eventuali equivocità, e potrebbe darci l'opportunità di risolvere l'ambiguità '*it*' distinguendo e connotando quali siano gli oggetti *storable* e quali non, oppure quali possano essere gli oggetti connotabili con l'attributo *veloci/lenti* e quali non.

Tuttavia se la premessa, ossia l'appartenenza del testo ad un dominio di corpora abbastanza specifici, venisse a mancare, la connotazione delle parole non sarebbe più sufficiente a risolvere l'ambiguità dato che anche il computer è un oggetto *storable*.

La semplice disambiguazione semantica necessaria per la scelta della parola giusta nella lingua target consiste nel caso specifico nella ricerca e scelta di "un candidato" fra i possibili per risolvere la referenza di *it*, e fra le due possibilità (*it* = *computer*/*computadora* oppure *it* = *the result/el resultado ?*), per poter scegliere è necessario avere conoscenze sufficientemente ampie sul dominio; ad esempio:

7. *John hit the dog with a stick* (*John hit the dog by a stick*) **à** *John golpeò el perro con el palo*, ma anche

8. *John hit the dog with a stick* (*that had a stick*) **à** *John golpeò el perro con el palo* (*que tenìa el palo*)

Questa ambiguità è risolvibile solo da contesto, e cioè se esso fornisce l'informazione che è John ad avere il bastone con il quale colpisce il cane, e non il cane ad avere, ad esempio, un bastone in bocca.

2.7.2 generazione del linguaggio: la scelta del tempo verbale

Un sistema di *M.T.*, fornendo una traduzione accettabile da un linguaggio *sorgente* ad uno *target*, deve essere in grado di selezionare ed usare le parole adeguate per la resa della traduzione.
Spesso sono gli stessi termini che contengono una connotazione sufficiente alla scelta della parola appropriata per il contesto da tradurre, ma altrettanto spesso, se le informazioni linguistiche non sono presenti o sufficienti nel linguaggio sorgente, come ad esempio nel caso in cui generare (e quindi decidere) il tempo verbale nel linguaggio finale, ad esempio:

9. *Cinese*: *Wǒ bèi Hàngzhōu de fēnjǐng xīyīnzhù le* **à** *Inglese*: *I was captivated by the scenery of Hangchow*, oppure *I am captivated by the scenery of Hangchow*.

In questo caso l'informazione necessaria alla selezione del tempo verbale adeguato dipende del tutto dal contesto del proferire: infatti la seconda traduzione sarebbe adeguata solo se chi parla sta osservando il paesaggio in quel momento.

2.7.3 relazioni fra coppie di lingue (linguaggio sorgente e linguaggio target)

Un problema linguistico ulteriore riguarda l'identificazione e la rappresentazione delle relazioni fra coppie di linguaggi, ma le divergenze che occorrono tra un linguaggio e l'altro rendono semplicemente impossibile eseguire una mappatura del tutto corretta tra coppie di lingue.

Analizzando la frase:

10.*Inglese*: *I like Mary* **à** *Spagnolo*: *Mary me gusta*

si evince che durante la traduzione è avvenuto uno scambio di posizione fra soggetto ed oggetto dall'Inglese allo Spagnolo (Ingl. Soggetto Predicato Oggetto **à** Sp. Oggetto Predicato Soggetto che, con questa costruzione tematica, avrebbe come risultato la frase: *Mary (to) me pleases*): questo fenomeno è provocato dalle divergenze tematiche fra le due lingue.

Un caso ancora più interessante e sottile è rappresentato da questo esempio:

11. *Inglese*: *I like to eat* **à** *Tedesco*: *Ich esse gern*,

dove il verbo *to like*, in Inglese il predicato principale della frase, in Tedesco viene trasformato ad un avverbio *gern*, e quindi il passaggio della frase fra le due lingue evince una divergenza strutturale: l'argomento del verbo del linguaggio sorgente ha una realizzazione sintattica differente nel linguaggio target.

Anche dalla frase:

12. *Inglese*: *John entered the house* **à** *Spagnolo*: *John entrò en la casa*

si evince una divergenza strutturale, poiché nella lingua inglese l'oggetto del predicato è reso da un sostantivo, *the house*, mentre in Spagnolo esso è introdotto da una preposizione, *en la casa*.

Altre divergenze riguardano la trasformazione da predicato aggettivale a predicato nominale, come in questo esempio:

13. *Inglese*: *I am hungry* **à** *Tedesco*: *Ich habe hunger*

in questo caso, addirittura, volendo tradurre inversamente, e cioè da Tedesco ad Inglese, il sistema di *M.T.* dovrebbe scegliere una voce verbale diversa, e la frase sarebbe resa con *I have hunger*, grammaticalmente scorretta.

Quest'ultimo esempio entra nella sfera del problema della traduzione di espressioni idiomatiche o *muliword expressions* che hanno (ma non sempre) corrispondenze al livello concettuale ma spesso hanno forme linguistiche molto diverse.

Un sistema di *M.T.* deve quindi avere una base di criteri con il quale costruire le corrette espressioni idiomatiche, e per evitare una costruzione grammaticale errata si compilano di solito dizionari specifici ai quali i *M.T.S.* si appoggiano per la generazione degli idiomi corretti nel linguaggio target.

Un ulteriore problemaconsiste anche nella traduzione del linguaggio "tecnico" relativo a determinati campi semantici, e per questo sono stati compilati dizionari terminologici specializzati e multilingue.

Differenze ancora più profonde generate durante la traduzione sono provocate dalle *divergenze conflazionali* fra coppie di lingue, dove cioè una proposizione espressa nella lingua sorgente ha bisogno di essere espansa per una resa grammaticalmente corretta nel linguaggio target: ad esempio:

14. *Inglese*: *I stabbed John* **à** *Spagnolo*: *Yo le di puñaladas a John*,

dove l'effetto dell'azione, in Inglese contenuto nel predicato, in Spagnolo deve essere specificato tramite il sostantivo *puñaladas*.

Riferendo l'esempio n alla traduzione dall'Inglese all'Italiano osserviamo un caso ancora più articolato:

15. *Inglese*: *Stabbed* **à** *Italiano*: *pugnalare* o
dare una pugnalata o
prendere una pugnalata o
prendere a pugnalate.

si tratta dell'uso dei così detti verbi di supporto, verbi con significato molto generico di azione seguiti da un sostantivo.

Alcune lingue, che non hanno il corrispettivo verbo possono ricorrere al posto di un verbo alla combinazione verbo di supporto + sostantivo (anche investire = fare un investimento).

La soluzione dei problemi dovuti alle divergenze fra coppie di lingue dipende molto anche dal modello e dalle risorse linguistiche utilizzati nei sistemi di traduzione automatica, dove, per fare solo un esempio, la costruzione di adeguate relazioni fra predicati ed argomenti da un linguaggio ad un altro risolvono alcune delle divergenze qui esposte.

Questo è il caso modello basato su una *Interlingua* (analizzeremo i modelli dei sistemi di *M.T.* successivamente), uno degli approcci più recenti, che sfrutta una rappresentazione dei significati indipendente dai linguaggi sorgente e target.

2.8 sistemi di M.T.: fattori operativi:

Oltre alle problematiche linguistiche precedentemente introdotte, un sistema di *M.T.* deve essere dotato di caratteristiche operative, quali:

- *l'estendibilità del sistema di M.T. alla manipolazione di altri linguaggi*;
- *la capacità di trattare diversi i stili testuali*;
- *la capacità di mantenimento una volta che esso sia stato sviluppato*;
- *l'integrazione con altre applicazioni*;
- *un metodo di valutazione per testare l'effettivo funzionamento del sistema*.

2.8.1 Estendibilità del sistema di M.T. alla manipolazione di altri linguaggi

Di solito un sistema di *M.T.* viene sviluppato su un dominio di testi ristretto, così la fase di comprensione da testi sorgenti e la fase di generazione dei testi target potrà avvalersi dell'uso di grammatiche e dizionari ad hoc (di dimensioni piu' ridotte) : attraverso questo sistema si restringono enormemente i casi linguistici a cui sopra abbiamo accennato.

Per poi estendere[40] il sistema ad un dominio più ampio, sarebbe necessaria l'acquisizione di nuovi termini da dizionari specializzati, ma sebbene siano stati sviluppati diversi approcci per l'acquisizione automatizzata di dati da lexicons, questi strumenti in realtà forniscono solo un supporto perché si limitano a ridurre il lavoro manuale che deve essere eseguito da personale estremamente specializzato sia in linguistica sia nel dominio al quale il dizionario ed il sistema di *M.T.* in upgrade appartengono

Le nuove parole che devono essere incluse nel sistema potrebbero essere estratte da un corpus annotato, ma comunque ognuna di queste nuove entrate dovrebbe essere riesaminata per controllarne l'accuratezza.

L'estensione del sistema di *M.T.* ad altri linguaggi deve inoltre prevedere la costruzione di un nuovo dizionario bilingue, un analizzatore grammaticale per ciascun nuovo linguaggio sorgente, ed una grammatica di generazione per il linguaggio target, e la creazione di queste nuove risorse richiede senza alcun dubbio l'intervento di personale linguisticamente specializzato, è un processo

[40] Upgrade del sistema.

lungo, laborioso e dunque costoso in termini di tempi e di risorse umane (*manpower*).

L'upgrade non dovrebbe essere limitato alla sola acquisizione di nuovi termini, ma dovrebbe riguardare anche l'analisi di strutture nuove: immaginando di possedere un sistema ben funzionale per la gestione della posta elettronica nel campo commerciale (lettere commerciali) ed ipotizzando di voler estendere lo stesso sitmea alla gestione delle notizie di Agenzia, in cui sono molto frequenti le espressioni ellittiche, oppure alla traduzione di manuali d'uso di apparecchiature varie, è evidente che, contemporaneamente all'estensione del dominio d'azione, il sistema dovrebbe essere aggiornato nelle proprie strutture per poter risultare efficiente.

2.8.2 la capacità di trattare diversi i stili testuali

un'altra importantissima caratteristica che il sistema deve possedere è la capacità di discernimento di quale sia lo stile del testo che sta esaminando, ad esempio: i testi letterari, così come quelli poetici e le novelle, usano frequentemente le metafore, la costruzione delle frasi è spesso complessa ed inusuale, e problemi simili si incontrano per trattando articoli di giornale, che oltre ai problemi precedentemente citati, spesso usano o creano parole il più "accattivanti" possibili e magari inusuali nel contesto di cui trattano: questi aspetti sono fuori della competenza dei sistemi di *M.T.* attuali. Essi sono invece applicabili a corpora di testi con sintassi semplice, che accingono ad un vocabolario stabile (e non in continua evoluzione come quello letterario o giornalistico), che fanno poco uso di metafore e che quelle usate siano facilmente comprensibili e dal dominio (di allusioni, sensi, ecc…) ristretto: questa categoria include testi tecnici e scientifici, ben trattati dalle attuali applicazioni di *M.T.*.

2.8.3 Capacità di mantenimento

Un'altra caratteristica propria dei sistemi di *M.T.* è quanto effettivamente possa costare mantenere un dizionario una volta che esso sia stato acquisito.

Le interfacce ideate per utenti, riguardanti però la categoria di traduzione assistita, dunque semi-automatica, senza requisiti linguistici sono invece principalmente organizzate con la presentazione di diversi tipi di frasi dove compare la parola da assumere, ed il sistema chiede all'utente di specificare quali siano corrette e quali non lo siano, ed ogni test rappresenta una discriminante per una particolare applicazione linguistica, tuttavia anche questo metodo, basato sulla costruzione di frasi, non da i risultati desiderati in tutti i casi: sarà quindi necessaria anche con questo tipo di software una revisione da parte di un traduttore.

2.8.4 Integrazione con altre applicazioni

Una caratteristica ulteriore che un sistema di *M.T.* dovrebbe avere è la possibilità di integrare il proprio operato ad altri software ed applicazioni, come il *riconoscimento ottico* dei *caratteri* (*O.C.R.*, *Optical Character Recognition*), e strumenti di *editor* e *pubblicazione* dei testi, e sempre di piu' con il riconoscimento e la generazione di parlato (*Speech*).

2.8.5 Metodo di valutazione per testare l'effettivo funzionamento del sistema

Una volta che le due applicazioni grammaticali, analizzatore sorgente e generazione target, sono state scritte, esse debbono essere seriamente testate fino a quando il loro risultato non sia accettabile su qualsiasi tipo di input venga inserito.

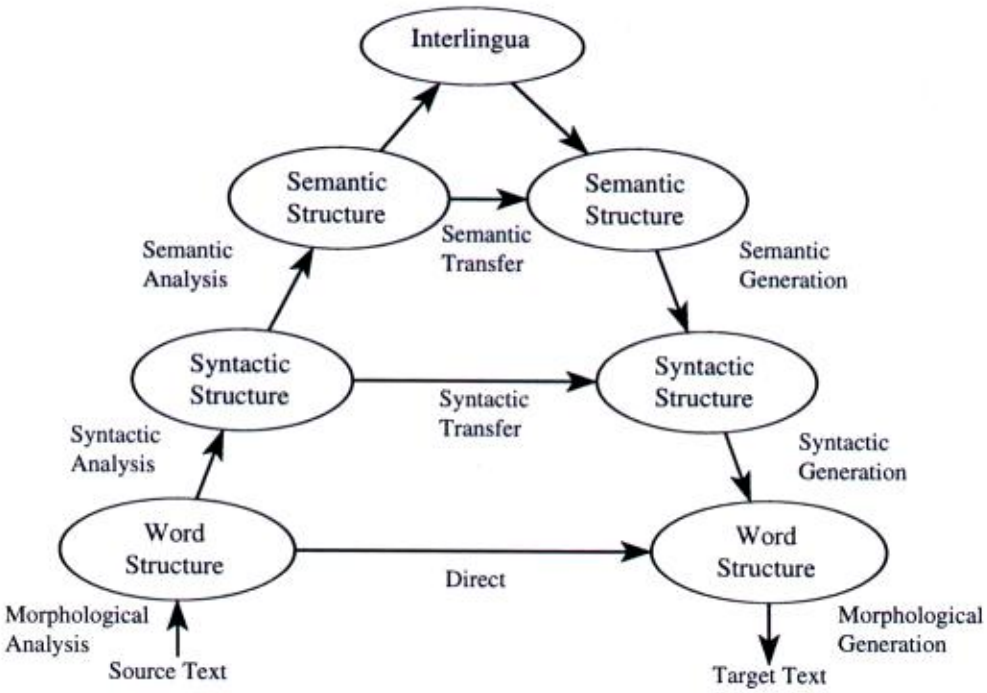
Sebbene questo aspetto faccia parte del processo di sviluppo di un sistema di *M.T.*, e questo sia uno delle fasi che richiedano molto tempo per un adeguato pestaggio del sistema, non sono ancora stati creati strumenti per automatizzare il procedimento.

2.9 I modelli dei sistemi di M.T.

Attualmente i modelli di tutti i sistemi di *M.T.* possono essere grosso modo raggruppate in tre classi:

1. *Modello diretto*
2. *Modello a Transfer*
3. *Modello ad Interlingua*

Questi tre livelli di modelli posso essere relazionati fra loro e rappresentati in un diagramma "a piramide" alla base della quale compare *l'approccio diretto*, che consiste nel metodo più primitivo di traduzione ed analisi, come ad esempio la sostituzione parola per parola.



Schema piramidale delle architetture di M.T.

Al vertice della piramide, invece, domina il sistema tramite *Interlingua*, che coincide con il metodo di traduzione più elaborato.

2.9.1 modello diretto

Il risultato di una traduzione tramite *M.T.* ad *modello diretto* si esplicita in una stringa dove ogni parola del linguaggio *sorgente* è sostituita dalla sua traduzione nel linguaggio *target.*

Riprendendo tutta la serie di problematiche linguistiche precedentemente incontrate, appare immediatamente un problema linguistico di tipo strutturale: generalmente l'ordine delle parole del linguaggio target differiscono da quello del linguaggio sorgente, fino a giungere al caso estremo dove il linguaggio target non permette la stessa costruzione del periodo.

Alcuni sistemi ad modello diretto, dunque, sono stati forniti di un'applicazione capace di riconoscere le forme sintattiche peculiari del linguaggio sorgente, e di riordinarle secondo forme accettabili nel linguaggio target.

Un esempio riproponibile è costituito dalla relazione tra verbo inglese *to like* e spagnolo *gustar*, rappresentabile nel seguente schema astratto: **X** *LIKE* **Y** **à** **Y** *GUSTAR* **X**.

L'esempio citato riguarda comunque un caso molto semplice ed intuitivo, una di quelle forme semplici che possono essere riconosciute agilmente, ma senza un analizzatore sintattico adeguato le forme dalle strutture più complesse, consequenziali, e tutte le regole ed i casi di separazione dei verbi (dove eccelle il Tedesco), non potrebbero essere riconosciute adeguatamente: in questi casi non

sarebbe possibile la costruzione di schemi di relazioni dirette fra coppie di lingue.

Altro serio problema dei sistemi di *M.T.* ad modello diretto consiste nel processo di disambiguazione dei termini per la scelta delle parole attinenti durante la generazione della stringa nel linguaggio target.

L'unica soluzione da adottare per far fronte alle difficoltà incontrate è la maggior restrizione possibile del dominio dei testi analizzati, così come la compilazione di un dizionario che contenga solo la traduzione più frequente per il dominio scelto, ma in ogni caso i sistemi di *M.T.* ad modello diretto applicati a testi comuni, producono, nel migliore dei casi, traduzioni povere.

Essi sono invece attendibili per corpora di testi molto specifici, dove il dominio dei corpora di testi è estremamente specifico, e quindi la restrizione del relativo dizionario provocata proprio dalla settorialità del linguaggio usato, sopperisce alle carenze lasciate dall'assenza di un analizzatore sintattico, e quindi produce risultati utili destinati, chiaramente, ad un'utenza esperta di tale dominio.

2.9.2 modello a transfer

Le architetture a *transfer* possono essere poste fra le architetture dirette e le architetture ad interlingua, e sono state concepite per fornire una rappresentazione sintatticamente corretta in un linguaggio target dalla sua rappresentazione nel linguaggio sorgente.

Sebbene le regole dell'interfaccia di *transfer* che permettono la traduzione dipendano da entrambi i linguaggi, a volte le regole sintattiche per l'upgrade del sistema di *M.T.* ad un nuovo linguaggio devono essere solo leggermente modificate.

Il sistema a *transfer* necessita di regole e di relazioni tra i due linguaggi, anche per la propria rappresentazione interna: in pratica deve essere costruito un sistema per la rappresentazione interna di ciascun linguaggio, ed una serie di regole di *transfer*, un'interfaccia, che permetta il link fra queste due rappresentazioni.

Il sistema di rappresentazione interna fra coppie di linguaggio sarà molto simile o identico (*SUBJ*, *OBJ1*, ecc…), e tentando di rappresentare l'esempio 9:

9. *I like Maria* **à** *Maria me gusta*, che risulterebbe

like(*SUBJ*(*ARG2*:*NP*),*OBJ1*(*ARG1*:*NP*)) **à** *gustar*(*SUBJ*(*ARG1*:*NP*), *OBJ*(*ARG2*:*PREP*))

L'effetto di questa regola di *transfer* è quello di scambiare il *soggetto* e *l'oggetto* del *predicato*, ed addirittura la *categoria degli oggetti*: da una *preposizione* (Spagnolo) ad un *nome* (Inglese).

Questo sistema permette inoltre di risolvere ambiguità lessicali sin quando l'analizzatore sintattico possa determinare la categoria grammaticale della parola del testo sorgente (*part of speech*): rimangono da risolvere le ambiguità lessicali e quelle sintattiche più difficili, ed in pratica frasi lunghe e complesse potrebbero risultare incomprensibili.

Per tentare di risolvere il problema, molti sistemi sono stati dotati di un analizzatore semantico e di regole appropriate, con il risultato di un'analisi combinata sintattico-semantica. Sebbene questa sia la soluzione adottata per tentare di risolvere alcuni problemi specifici tra le coppie di lingue, l'analisi semantica rimane incompleta e, in qualche caso, dipendente dalla coppia di lingue.

Nei migliori casi i sistemi di *M.T.* con modello a *transfer* producono traduzioni eccellenti, perché prodotte tramite analisi sintattica e da una profonda analisi semantica, risultato sicuramente di livello superiore rispetto a quello raggiungibile tramite sistemi ad modello diretto, ma con il peso di dover implementare tecniche di analisi estensive del linguaggio sorgente e serie di regole di *transfer* per ciascuna lingua.

Un esempio di tale sistema è il *SYSTRAN* [41] usato dalla *Comunità Europea*.

2.9.3 modello ad interlingua

L'idea basilare di un sistema con modello ad interlingua per applicazioni di *M.T.* consiste nel fatto che l'analisi del testo nel *linguaggio sorgente* deve risultare in una rappresentazione del testo indipendente dal *linguaggio sorgente* stesso: il *testo target* sarebbe dunque poi generato da questo testo in "*linguaggio neutro*".

Attualmente vi sono già dei sistemi commerciali che hanno adottato questo approccio, e questa è una delle più attive aree di ricerca: gli sforzi sono orientati sulla possibilità di generare *un'interlingua* adeguata per tutti i linguaggi, che sulla base di analisi semantiche, riesca a fornire delle traduzioni accettabili.

Applicando dunque questo principio teorico sempre all'esempio 9, il sistema interlingua dovrebbe assumere l'esistenza di un concetto per il significato dei verbi delle due frasi, e la rappresentazione dell'esempio *like* **à** *gustar* potrebbe essere la seguente:[42]

like/*gustar*: [*CAUSE*(*X*,[*BE*(*Y*, [*PLEASED*])])]

[41] *Systran* sarà approfondito successivamente.

[42] Ci sono tantissimi altri modi di rappresentare concetti in un'interlingua, per esempio ontologie, strutture tassonomiche, ecc. L'interlingua fa di solito uso di un dizionario concettuale, da cui poi creare i dizionari specifici per ogni lingua. La differenza è che, invece di passare da PAROLA in linguaX **à** PAROLA in lingua Y si passa da
CONCETTO in interlingua **à** PAROLA in lingua X, PAROLA in lingua Y, ecc…

Questa rappresentazione comunica l'idea che qualcuno o qualcosa (X) causi (Y) ad essere felice.

La situazione attuale sembra indicare che sia possibile produrre delle *interlingue* adeguate fra gruppi di linguaggi (ad esempio Europeo e Giapponese) il cui uso sia ristretto a particolari domini.

3 Sperimentazione di biblioteca digitale per l'Egittologia con accesso intelligente e multilinguistico alle informazioni

3.1 utilizzazione di sistemi di MLIA e CLIR ad applicativi egittologici

La costruzione di una biblioteca digitale dedicata ad un tema egittologico avrebbe potuto essere realizzata in diversi modi, ciascuno dei quali avrebbe permesso l'utilizzo di strumenti sempre migliori e più precisi per una consultazione della stessa bibliografia elettronica sempre più soddisfacente da parte dell'utenza.

I punti salienti, se correttamente implementati ed approfonditi che avrebbero garantito la produzione di un archivio digitale ben consultabile sarebbero stati essenzialmente due, apparentemente separati, ma in realtà in stretto rapporto fra loro, ossia:

1. La realizzazione dei sistemi di ricerca, strumenti essenziali per l'interrogazione ed il reperimento delle informazioni richieste (realizzazione dell'Information Retrival System)

2. La realizzazione delle parti multilinguistiche del portale (applicazione dei criteri e degli strumenti di M.L.I.A.)

1. Il primo punto, ossia la realizzazione dei sistemi opportuni alla ricerca avrebbe comportato prima di tutto la creazione dell'interfaccia di comunicazione e presentazione all'utente e, soprattutto, la realizzazione di quelle strutture necessarie all'interrogazione verso il database.

In questa fase sarebbe dunque stato necessario rispettare e mantenere i criteri di *efficienza* ed *efficacia* all'interno del sistema di *Information Retrival System*,[43] necessità che avrebbero comportato la rielaborazione per lo meno di due altre parti strutturali del portale *Bibliografiapiste*:

- la costruzione di indici adatti con i quali classificare e rendere reperibili i singoli documenti alle singole *query*

- la costruzione di tutti quegli applicativi necessari alla manipolazione delle stringhe di *query* fornite dall'utente in fase d'interrogazione, riassumibili nei tre processi di *tokenizzazione*, eliminazione delle *stopwords* e *normalizzazione* della *query*

[43] Per quanto concerne i criteri enunciati si rimanda a *2.3 Sistemi di Information Retrival*.

2. Per quanto concerne invece la realizzazione delle parti *multilinguistiche*, questo processo sembrerebbe apparentemente legato alle sole fasi "*iniziali*" e "*finali*" del processo di interrogazione verso l'archivio digitale.

Sembrerebbe cioè lecito ipotizzare che per la realizzazione di questo punto sarebbe stato semplicemente necessario costruire una "versione" in più lingue delle sole strutture d'interfaccia del portale Bibliografiapiste, riducendo così il lavoro ad un' operazione più o meno compilativi, le "traduzioni" necessarie, per poter costituire uno strumento di MLIA e rendere così il portale ad una versione multilinguistica.

Questa soluzione sarebbe stata limitata ad un primo livello di approccio verso un sistema di accesso ad informazioni multilinguistico (I *livello di MLIA*), mentre l'applicazione del multilinguismo sia alla fase di realizzazione del sistema di ricerca (II *livello di MLIA*) sia alla visualizzazione dei documenti contenuti nell'archivio digitale visualizzati alla conclusione di un'interrogazione (III *livello di MLIA*) avrebbe consentito la realizzazione del portale Bibliografiapiste multilinguistico su due ulteriori e maggiori livelli di MLIA:

> I *livello di MLIA*: questo livello applicativo è il più semplice e più diffuso, e come abbiamo già accennato, si limita alla costruzione delle singole interfacce di dialogo e di presentazione delle pagine del portale in più lingue.
> Nel caso di Bibliografiapiste abbiamo la realizzazione di quattro versioni: Italiano (ovviamente) Inglese, Spagnolo e Portoghese[44]
> Questo sistema garantisce ad un qualsiasi utente di scegliere una delle lingue supportate da un menù iniziale e quindi di apprendere circa gli argomenti trattai all'interno del portale, di capire il tema egittologico specifico a cui è dedicata e raccolta la bibliografia fornita, e dunque garantisce l'accesso nella lingua prescelta a tutti gli strumenti necessari alla consultazione dell'archivio digitale.
> Essendo questo solo un primo livello di approccio multilinguistico, sono evidenti due principali:
>
> 1. prima di tutto malgrado l'utente avesse accesso ad un'interfaccia nella lingua a lui più confacente, egli non avrebbe potuto utilizzare la lingua prescelta per formularvi query, dato che il linguaggio supportato da tale strutture non prevedeva ancora varietà multilinguistica; l'eliminazione di questo

44

limite avrebbe costituito un livello ben diverso e più profondo di M.L.I.A..

2. Altra carenza evidente consisteva nel fatto che i documenti relativi ad una *query* (a prescindere dal fatto che il primo punto, ossia II *livello di MLIA* fosse stato implementato o meno) sarebbero stati mostrati all'utente nella lingua con cui essi erano stati previamente archiviati all'interno del database, precludendo quindi l'accesso alle informazioni rappresentate da ciascun testo in modo *multilinguistico*.

II *livello di MLIA*: per ovviare al primo e forse più interessante dei limiti incontrati sarebbe stato necessario implementare un sistema di C.L.I.R., architettando cioè un sistema di Information Retrival tale da supportare durante la fase di interrogazione più linguaggi, per lo meno gli stessi con i quali si era deciso di le pagine durante la realizzazione del primo livello di MLIA, ossia Italiano, Inglese, Spagnolo e Portoghese.
L'implementazione di queste strutture avrebbe costituito il *secondo livello di MLIA* all'interno del portale Bibliografiapiste, poichè a questo punto un utente avrebbe potuto scegliere fra i linguaggi disponibili il più opportuno alle proprie necessità, ed attraverso le interfacce di ricerca e di presentazione fornite nella lingua scelta, avrebbe potuto interrogare il database con query espresse questa volta nella lingua inizialmente prescelta, riuscendo ugualmente nella corretta estrazione dei documenti pertinenti a prescindere da quale fosse l'originale contenuto nel database.
Nel portale Bibliografiapiste questo livello ulteriore di MLIA è stato raggiunto indicizzando i testi rappresentati nell'archivio digitale attraverso un'ontologia (INFOR) e tramite la rielaborazione delle stringhe di query attraverso processi di Stemming.
I documenti relativi ad una query, malgrado la loro corretta estrazione indipendente dal linguaggio dell'interrogazione, sarebbero stati comunque visualizzati sempre nella lingua con cui essi erano stati previamente archiviati all'interno del database.

III *livello di MLIA*: Il III livello di *MLIA* avrebbe cercato di ovviare a quest'ultima carenza, nel tentativo di produrre, una

volta estratti i documenti relativi ad una certa query, formulata in una delle lingue supportate dal sistema, le schede ad essi relativi tradotte nella lingua prescelta inizialmente dall'utente.
Egli sarebbe stato in grado cioè, a prescindere dai linguaggi originali dei testi archiviati all'interno della biblioteca digitale, di scegliere, interrogare e leggere i risultati delle proprie query nella lingua preferita.
Questo avrebbe consistito nella realizzazione del *III livello di MLIA* all'interno del portale Bibliografiapiste, e per la sua realizzazione è stato utilizzato il sistema di Machine Translation Babelfish/Systran.
Vi è inoltre affrontato un tentativo ulteriore di approccio alla questione, prendendo in considerazione l'utilizzo di un altro sistema di Machine Translation, U. N. L., le cui peculiari caratteristiche hanno permesso anche la teorizzazione e la discussione di un ulteriore prototipo di C.L.I.R.s..

3.2 I livello di MLIA: costruzione del portale bibliografiapiste e del databe ospite dei testi

3.2.1 definizione delle strutture da implementare

La precisa definizione di come strutturare la risorsa bibliografica, cioè la stesura di un progetto chiaro, avrebbe costituito la necessaria premessa per la realizzazione della bibliografia digitale.

Il particolare tema egittologico, così ricco di diversi tipi di documenti, ha rappresentato la prima difficoltà di natura strutturale della bibliografia stessa: questa infatti avrebbe contemplato una svariata tipologia di testi come libri, articoli tratti da riviste e periodici, contributi ed atti di convegni e congressi, testi geroglifici, materiale fotografico di svariato genere, cartine topografiche, foto, ecc...

Tutta questa serie di documenti avrebbe dovuto essere catalogata in modo da creare due strutture: una *bibliografia topografica* ed un archivio *OPAC* , quest'ultimo di dimensioni più contenute possibili, ma al tempo stesso connotato da criteri che avrebbero poi rappresentato le chiavi per la reperibilità dei documenti una vota pubblicato il tutto sul web.

Il portale avrebbe allora dovuto contenere due sezioni principali:

- *bibliografia topografica*: ossia l'organizzazione degli argomenti rappresentati dalla bibliografia raccolta per indice topografico, e quindi riportando una lista delle piste carovaniere analizzate dove per ciascuna di esse si fornisse la bibliografia specifica,

- *archivio bibliografico OPAC*: e cioè un archivio bibliografico che disponesse i documenti della bibliografia in modo classico, organizzati e rappresentati secondo delle schede bibliografiche ordinabili secondo alcuni criteri e reperibili tramite interrogazioni da un motore di ricerca.

3.2.2 Progettazione della bibliografia topografica

La realizzazione della *bibliografia topografica* avrebbe comportato la costruzione di un indice, la cui struttura si sarebbe articolata su tante voci quante sarebbero state le piste carovaniere da visualizzare, e per ciascuna di queste voci sarebbe stato necessario illustrarne la bibliografia relativa tramite una scheda appropriata; all'interno di ciascuna scheda, dopo la citazione della bibliografia specifica, sarebbe seguita un breve descrizione della pista carovaniera stessa desunta dalla consultazione e dallo studio dei documenti citati.

Ogni scheda della *bibliografia topografica* sarebbe stata strutturata in questo modo:

Toponimo :	Pista carovaniera da…a…
Bibliografia :	*Testo 1*: da…a… *Testo 2*: da…a… Ecc…
Descrizione :	La pista carovaniera da…a… e

3.2.3 Progettazione dell'archivio OPAC

Per quanto concerne, invece, la costruzione dell'archivio digitale *OPAC*, la struttura sarebbe risultata molto più articolata.

Sarebbe stato necessario, prima di tutto, costruire uno schedario elettronico che non andasse più a rappresentare un argomento (pista carovaniera) e la sua relativa bibliografia (citazione di parte di testi, articoli, ecc…), ma vi era la necessità di costruire un archivio canonico di schede, atto alla descrizione dei testi di una "normale" biblioteca elettronica.

Mentre cioè per la struttura precedente sarebbe stato sufficiente compilare una pagina che raccogliesse tutte le citazioni bibliografiche specificando tutti gli estremi relativi a ciascun testo o alla parte di esso dedicata alla singola voce (il caso maggiormente frequente), e quindi la scheda di ciascun elemento della bibliografia topografica avrebbe rappresentato un toponimo con relativa bibliografia, la scheda di ogni documento dell'*OPAC* avrebbe dovuto contenere voci distinte e necessarie alla descrizione completa di ciascun documento, dato che quest'ultimo sarebbe stato l'oggetto da rappresentare.

Per ciascun documento, quindi, doveva essere chiarito di che "*tipo*" fosse: esso era un libro, oppure un articolo tratto da un periodico, oppure un contributo o un atto di congresso, ecc…

Dopo aver chiarito la "*natura*" del documento, era necessario fornirne l'autore, il titolo, se un libro il codice *ISBN*, se una rivista il codice *ISSN*, la data di pubblicazione e l'editore.

Essendo quindi lo scopo della pubblicazione sul web quello di procedere oltre la semplice lista di pubblicazioni, e fornendo il maggior apporto d'informazioni possibili per ciascun documento, apparve necessario:

- Citare l'argomento o gli argomenti principali presenti nel testo, e quindi il soggetto o i soggetti trattati dal documento

- Essendo l'argomento le piste carovaniere del deserto occidentale anche di carattere topografico, specificare

tutti i toponimi ed i soggetti topografici di cui ogni documento avesse trattato, o da cui esso provenisse

- Fornire un sunto del contenuto del testo archiviato, oppure di una parte di esso

Ogni "*scheda bibliografica*", dunque, avrebbe dovuto consistere in:

Tipo documento :	Libro, Articolo, Atti di Congresso, Contributi, altro…
ISBN / ISSN :	Il numero identificativo se libro o periodico
Autore :	Cognome e nome autore/i
Titolo :	Titolo del documento, libro, ecc…
Edito :	Editore, anno d'edizione
Soggetto :	Soggetto/i di cui parla il documento
Soggetto Topografico :	Indice dei Toponimi e soggetto/i topografici
Riassunto :	Breve sunto del documento o della parte di esso consi

L'ultimo campo, il *riassunto*, avrebbe forse costituito il punto più interessante per questa applicazione, perché esso avrebbe fornito al potenziale utente una risorsa avanzata per la sua ricerca, mentre troppo spesso le applicazioni bibliografiche online si limitano a fornire una lista di documenti attinenti ai criteri predefiniti (ad esempio mostrare tutti i documenti pubblicati in un determinato anno) od immessi dall'utente (ad esempio ricerca di una *query* fra i documenti di un archivio).

Il *soggetto topografico* ed il *riassunto* avrebbero fornito all'utenza una miglior fonte per chiarire meglio se i testi apparsi come risultato da una ricerca, fossero effettivamente attinenti alle proprie esigenze, quali i più adatti, quali avessero presentato i contenuti individuati perché effettivamente argomento del testo, o magari perché appena accennati al suo interno.

Sarebbe stato opportuno poter rendere il contenuto del campo *riassunto* in più lingue, questo nell'ordine di fornire un servizio di *Information Retrival* il più completo possibile, nell'ordine di sviluppare un *MultiLingual Information Acces* completo sotto tutti i suoi aspetti.

Lo sviluppo della biblioteca digitale si sarebbe arricchita di una caratteristica importantissima: essa sarebbe divenuta una risorsa multilinguistica, riprendendo dunque tutti i concetti precedentemente descritti nella 2° parte dedicata all'*Information Retrival*, e la bibliografia multilinguistica, una volta pubblicata sul web avrebbe dovuto dialogare in più lingue

- *Nell'interfaccia utente* (la parte di presentazione e di descrizione del sito web, i suoi menù, le "istruzioni" per l'uso adeguato della bibliografia, ecc…)

- *Nella "scheda" bibliografica*, fornendo cioè la descrizione di ciascun libro o documento archiviato nella bibliografia.

Considerando le premesse esposte nella sezione dedicata all'Information Retrival, a cui già si è fatto riferimento precedentemente, sarebbe stato necessario scegliere quali lingue avrebbero dovuto essere rappresentate nella bibliografia digitale, e direttamente in relazione proprio a questa scelta, di quali supporti informatici adottare per la sua realizzazione.

Ovviamente sarebbe stato preferibile quel supporto che avesse permesso la traduzione dei testi in quante più lingue possibili, coprendo immediatamente almeno la lingua Inglese e Spagnola, ed essendo aggiornabile successivamente ad altre lingue.

La realizzazione del primo punto, l'interfaccia utente, si sarebbe limitata nella traduzione delle parti scritte delle pagine web: i suoi menù, le istruzioni per l'uso e la consultazione della bibliografia topografica e del catalogo OPAC, e quindi avrebbe dovuto consistere in un lavoro relativamente impegnativo e più o meno non suscettibile ad ulteriori analisi.

La seconda parte necessitava di più attente riflessioni. infatti , mantenendosi attinenti al modello di scheda bibliografica costruita, sarebbe stato evidente che non tutti i campi di essa sono da rendere in più traduzioni, ma anzi per una puntuale descrizione del documento analizzato, i campi come *Autore*, *Titolo*, ovviamente *ISBN / ISSN*, ed *Edizione* avrebbero dovuto essere mantenute nella lingua della redazione originale del testo archiviato, e dunque specificando i campi della scheda bibliografica dal punto di vista linguistico essa avrebbe assunto questo aspetto:

Tipo documento :	*Multilingua*
ISBN / ISSN :	Alfanumerico
Autore :	Lingua originale
Titolo :	Lingua originale
Edito :	Lingua originale
Soggetto :	*Multilingua*
Soggetto Topografico :	Lingua originale
Riassunto :	*Multilingua*

I campi i cui contenuti avrebbero dovuto essere resi in più traduzioni risultavano ridotti a 3: il *Tipo di documento*, il *Soggetto* ed il *Riassunto* del testo.

Questa struttura avrebbe dunque accolto, classificato e mostrato l'intera bibliografia con l'opportunità per l'utente di scegliere la lingua che più si addiceva.

La tabella che avrebbe accolto la bibliografia, ossia la struttura più articolata e più complessa dell'archivio bibliografico era stata ormai progettata completamente.

La possibilità di consultazione di una bibliografia topografica avrebbe costituito uno strumento di ricerca veramente soddisfacente da parte di quanti avessero intrapreso un'interrogazione alla bibliografia digitale,

A questo punto sarebbe stato possibile già delineare una possibile struttura finale del sito web:

A. *Home page* con titolo e menù per accedere alla bibliografia topografica o alla sezione OPAC. Entrambe le sezioni avrebbero contenuto inoltre la possibilità di visualizzarne l'interfaccia in Italiano od in Inglese,

B. *Sezione Bibliografia Topografica*, contenente una lista di tutte le voci della bibliografia, praticamente consistenti nelle piste carovaniere identificate ed analizzate tramite la bibliografia stessa. Ad ogni voce avrebbe corrisposto una pagina contenente la bibliografia specifica e la descrizione.

B1. *Pagina introduttiva alla sezione OPAC* ed agli argomenti trattati. La pagina avrebbe dovuto costituire la schermata di "benvenuto" all'utente, descrivere le risorse disponibili e contenere il menù con le voci per poter dare accesso all'utente alle diverse risorse del sito, e quindi:

B2. *Articoli ed estratti*: sezione dedicata alla consultazione dei periodici e degli atti. La pagina avrebbe permesso all'utente una consultazione mirata a questo tipo di testi velocizzando e facilitando la ricerca dell'utente. Tutti i documenti sarebbero stati ordinati per titolo.

B3. *Ordinamento di tutti i testi*, la pagina avrebbe contenuto una breve descrizione dei criteri con cui i documenti si trovano classificati nella bibliografia digitale, e da essa l'utente avrebbe potuto visualizzare tutti l'archivio ordinato secondo diversi criteri predefiniti:

- *per autore*,
- *per titolo*
- *per Isbn o Issn*
- *per tipo di documenti*
- *per soggetto*
- *per toponimi*
- *per data di pubblicazione*

B4. *Ricerca per campi*: l'utente avrebbe potuto interrogare l'archivio con una stringa da lui digitata (*query*). L'interrogazione verso l'archivio sarebbe rimasta vincolata a determinati campi della scheda bibliografica. era necessaria una pagina web che contenesse i campi di ciascuna scheda bibliografica sotto cui andare a cercare l'interrogazione proposta dall'utenza. L'utente inoltre avrebbe dovuto poter compilare e cercare all'interno dell'archivio più parole su più campi (eseguire cioè una ricerca incrociata) ed il motore di ricerca avrebbe dovuto restituire solo quei documenti che corrispondessero a tutti i criteri e le opzioni di ricerca scelte dalla pagina Ricerca per campi. I campi avrebbero dovuto essere i seguenti:

- *Autore*
- *Titolo*
- *Soggetto*
- *Topografia*

Questi quattro campi avrebbero permesso il recupero di qualsiasi testo da parte di quanti conoscessero le specifiche di almeno uno di essi.

B5. *Ricerca libera*, la ricerca non sarebbe stata vincolata a nessuno dei campi in questione, ma avrebbe puntato al "contenuto" di ciascun testo.
Sarebbe stato necessario fornire una pagina dalla quale l'utente potesse interrogare senza restrizioni su tutti i

campi di tutto l'archivio, o comunque sui contenuti di ciascun documento archiviato.
Anche in questo caso sarebbe stato indispensabile, però, distinguere delle opzioni: infatti se la *query* fosse stata composta da più parole (il caso più frequente), l'utente avrebbe potuto scegliere se cercare:

- *tutte le parole digitate*: i documenti avrebbero dovuto contenere, anche se non consecutivamente, tutte le parole contenute nella *query* di ricerca
- *almeno una delle parole digitate*: caso in cui i documenti avrebbero dovuto corrispondere almeno per una delle parole componenti l'interrogazione
- *la frase esatta*: il motore di ricerca avrebbe dovuto estrarre quei documenti che contenessero esattamente l'interrogazione digitata dal form di ricerca.

La *ricerca libera*, doveva quindi essere organizzata in modo che andasse ad interrogare una parte invisibile all'utente, ma in realtà presente, dell'archivio bibliografico contenente una sorta di rappresentazione astratta dell'argomento del testo.
Sarebbe dunque stato necessario, a prescindere dall'implementazione del sistema, aggiungere alla struttura di scheda bibliografica progettata un campo ulteriore che contenesse questa rappresentazione. [45]
Altra caratteristica necessaria sarebbe stata la realizzazione della *ricerca libera* applicandovi un sistema di *Cross-Language Information Retrival*, in base a quanto enunciato alla già citata 2° parte riguardante l'*Information Retrival*.

Il progetto si era dunque concluso con la precisa definizione delle strutture necessarie ad archiviare i testi da rendere fruibili sul web, e con l'identificazione delle pagine necessarie allo sviluppo completo del sito.

3.2.4 la scelta dei supporti informatici

[45] La ricerca per *parola chiave/libera* sarà affrontata successivamente in modo esauriente.

La preventiva e precisa definizione della struttura bibliografica e delle pagine che avrebbero costituito il sito web ha consentito una maggior precisione nella scelta dei supporti informatici.

La struttura della *bibliografia topografica* non avrebbe richiesto una particolare modalità di archiviazione dei propri contenuti, ma sarebbe stato sufficiente costruire due pagine, contenenti rispettivamente la versione in *Italiano* ed *Inglese* di tutte le voci della *bibliografia topografica*, e dunque per ciascuna di essa una pagina relativa contenete la bibliografia specifica e la descrizione.

La sezione *OPAC*, come progettato, avrebbe invece avuto bisogno di un archivio digitale, *un database*, contenente la *bibliografia* all'interno del quale avrebbero dovuto essere trasferiti i testi.

Ogni elemento del *database*, detto *tabella*, avrebbe contenuto le strutture necessarie alla archiviazione dei singoli documenti, e dunque sarebbe stato suddiviso in righe dette *campi*, costruiti come precedentemente accennato, ad immagine delle strutture cartacee precedentemente progettate.

MySql sarebbe stato il *manager* più efficace per la realizzazione del database[46] e scelto il supporto del database, sarebbe però stato necessario creare due interfacce che dialogassero con *MySql*:

- *L'interfaccia per l'amministrazione del database*, munita dunque di una serie di strumenti necessari all'aggiunta di nuovi record, alla modifica, alla cancellazione dei dati contenuti, all'amministrazione e la gestione di tutta la bibliografia, ecc...

- *L'interfaccia utente*, ossia quella serie di pagine e di form che dovevano permettere la navigazione all'interno del database per l'esecuzione di una ricerca e per la visualizzazione dei dati estratti dall'archivio, da realizzare inoltre in più lingue.

Entrambi i punti rimanevano condizionati da una scelta precedente: con quale strumento informatico realizzare il "dialogo" tra l'utente ed il database tramite cui realizzare gli strumenti di ricerca e comunque di manipolazione necessari sia all'utente sia all'amministratore rispettivamente per una consultazione ed una manutenzione ottimale dei dati contenuti nel *database*.

La risposta si sarebbe trovata proprio in una delle caratteristiche di *MySql*: la propria permeabilità al linguaggio *Php.*[47]

---

[46] Per le motivazioni ed approfondimenti relativi alla scelta di determinati applicativi si rimanda all'appendice Manuale Amministrazione **à** *1. La scelta dei supporti informatici*

[47] Anche in questo caso si rimanda all'appendice Manuale Amministrazione **à** *1. La scelta dei supporti informatici.*

L'ultimo supporto informatico, ma non ultimo per importanza visto che senza di esso nulla avrebbe potuto essere pubblicato online, sarebbe stato lo spazio web necessario per accogliere, conservare e rendere fruibile sul web sia il *Database* contenente la bibliografia sia le pagine dinamiche ed *Html* per l'interfaccia utente ed amministrazione.

Si rendeva necessario che il *Server* che gestisse questo spazio web supportasse (e cioè avesse i programmi interpreti necessari) *Php* e *MySql*: la scelta, quasi obbligata, è caduta sul server *Apache*.[48]

3.3 la realizzazione

3.3.1 costruzione della bibliografia topografica

Come da progetto, la costruzione della *bibliografia topografica* avrebbe dovuto consistere nella realizzazione di due principali tipi di pagine:

- *la realizzazione dell'indice della bibliografia*: organizzando le voci consultabili, specificando per ciascuna delle piste il sito di partenza, generalmente corrispondente ad un'oasi del deserto occidentale egiziano, o comunque ad un sito all'interno della depressione di una delle oasi, e quindi la destinazione, per poi raggrupparle secondo l'oasi di partenza in una sorta di albero i cui nodi sarebbero stati costituiti dalle oasi o dai toponimi di partenza, ed i rami le piste carovaniere che ivi partivano,

- *la realizzazione, per ciascuna delle voci della bibliografia topografica*, di una pagina contenente la bibliografia specifica relativa ed una descrizione della pista stessa e dei siti incontrati durante il proprio percorso. Dopo le opportune citazioni, la scheda avrebbe contenuto la descrizione dedotta dallo studio della bibliografia citata.

Entrambe le strutture avrebbero dovuto avere la versione corrispondente in *Inglese*, la cui realizzazione avrebbe usufruito del traduttore automatico *BabelFish*,[49] e sarebbero state tutte realizzate in *HTML* con un semplice sistema di links ipertestuali.

3.3.2 la costruzione dell'OPAC: il database Bibliografiapiste

[48] Vedi nota precedente.
[49] Vedi la parte dedicata agli applicativi di *Machine Translation*.

Dopo aver istallato e configurato correttamente *PhpMyAdmin* sul *Server*[50] è stato possibile creare il database che avrebbe accolto le tabelle precedentemente progettate, chiamato quindi *bibliografiapiste* che avrebbe contenuto la tabella *libri*, preparata per archiviare i testi della bibliografia digitale

Il database sarebbe dunque stato pronto ad archiviare i diversi documenti costituenti la bibliografia, e tramite le strutture successivamente implementate avrebbe reso possibile la fruizione della bibliografia digitale sul web.

Ovviamente vi sarebbero stati due livelli d'accesso ai dati ed alle strutture del database *bibliografiapiste* corrispondenti a due diverse gerarchie: un *superutente*, capace di lettura e scrittura su tutti i dati e le architetture del *database*, ed un accesso *utente*, destinato al dialogo standard con la *bibliografia*, capace dunque di visualizzare i valori (solo alcuni) contenuti nelle diverse tabelle, ma privo dei permessi di modifica.

3.3.3 costruzione della tabella libri

La costruzione della tabella per la bibliografia consiste nella dichiarazione del numero di campi necessari[51] e delle variabili contenute nei campi[52]

Essa consisteva di 11 campi, uno in più di quelli progettati, e cioè il campo id, un identificatore numerico per fornire all'amministrazione un ulteriore strumento per identificare i record:[53]

---

[50] Per la corretta configurazione di *PhpMyAdmin*, così come per comprendere l'istallazione del programma sul *Server* consultare i links forniti in nota 12. Esistono anche altre guide più o meno precise dedicate alla configurazione ed all'upload di *PhpMyAdmin* sul *server*, reperibili su portali che offrono servizi di web *Hosting*, come *Lycos* (http://www.lycos.it oppure http://www.tripod.lycos.it/) che amministra i database degli spazi web dei propri utenti proprio tramite *PhpMyAdmin* configurato ovviamente con livelli di accesso limitati.

[51] righe della tabella

[52] colonne della tabella

[53] Da sottolineare l'ampia flessibilità dei tipi di variabili che offre MySql, ad esempio nel caso dei campi che conterranno testo: già dalla dichiarazione dei tipi abbiamo una connotazione del contenuto dei campi stessi.
Il campo *id* conterrà un numero intero, ed è sostanzialmente un indice per una rapida selezione dei record. La specifica (11) definisce la lunghezza massima delle cifre del numero, così da allocare lo spazio di memoria disponibile per ciascun record in modo preciso ed efficiente.
Il campo Type, come precedentemente illustrato, conterrà il tipo di documento archiviato. Esso conterrà quindi un testo, definito come tinytext poiché la sua descrizione è esigua. Inizialmente avevo progettato che questo campo fosse con 4 valori fissi, contenenti i principali tipi di documento che prevedevo di prendere in esame per la realizzazione della mia tesi di laurea. Nel menù di amministrazione della bibliografia avrei fornito quindi le 4 opzioni da poter scegliere:

- Libro (o Book),
- Rivista (Periodic),
- Contributo – Atti Congresso (Tributes - Congress),
- Altro (Other)

Id :	*Iint(11)*
Type :	*Tinytext*
Author :	*Text*
Title :	*Text*
isbn/issn :	*Bigint(20)*
Publisher :	*Text*
Published :	*year(4)*
Subject :	*Text*
Topography :	*Text*
Abstract :	*Longtext*

Come già espresso nella fase di progettazione, questa struttura avrebbe descritto dettagliatamente ed esaurientemente qualsiasi testo della bibliografia digitale.

Avendo poi specificato praticamente tutte le "caratteristiche" di qualsiasi documento cartaceo, sarebbe stato possibile manipolarne i *record* contenuti toccando qualsiasi suo aspetto, potendo così fornire all'utente la possibilità di ordinare e cercare con precisione ogni testo per ogni suo aspetto: ad esempio sarebbe stato possibile ordinare tutti i documenti archiviati secondo il loro codice identificativo (*isbn* o *issn*), per autore, per titolo, ecc…, oppure eseguire una ricerca più mirata che, tramite appositi form, avrebbe permesso di cercare, ad esempio, un autore o un titolo in particolare oppure avendo più dati o

---

Il campo *type* allora sarebbe stato dichiarato come *varchar*(1), e le 4 opzioni sarebbero state specificate a *PhpMyAdmin* in fase di dichiarazione della variabile del campo type: type avrebbe assunto come valore il primo carattere di ogni voce (se type = libro => type = 'L', else se type = Rivista => type = 'R', ecc…).
Questa soluzione era però poco efficiente perché impediva una maggior precisione nello specificare alcuni tipi di documenti di varia tipologia, il cui "nome" non era semplificabile a Libro, o rivista, ma dove occorrevano ulteriori notazioni, e quindi sarebbe stato necessario creare altre voci per archiviare testi geroglifici, iscrizioni, tutto il materiale fotografico (tra l'altro non generalizzabile sotto un'unica dicitura perché di varia natura, ad es. rilievi topografici fotografie aeree, fotografie satellitari, ecc…), andando quindi a compromettere:

- *l'economia* del campo, poiché si dovevano costruire tutte questi ulteriori "tipi" di testo,
- la *precisione* con cui archiviare, e quindi poi pubblicare la bibliografia stessa, caratteristica forse più importante.

Optare dunque per realizzare il campo *type* come *tinytext*, ha concretizzato la possibilità di descrivere esaurientemente la tipologia del documento da archiviare, rispettando però i criteri di efficienza necessari al corretto funzionamento del database.

desiderando testi più specifici, interrogando l'archivio bibliografico tramite una ricerca incrociata.[54]

Strumenti essenziali per una connotazione dettagliata di ciascun testo sono i campi *topography*, *subject* ed *abstract*:

- Il campo *topography* avrebbe costituito la risorsa tramite cui realizzare una bibliografia topografica, attraverso la quale sarebbe stato possibile visualizzare tutti i record della tabella libri, in ordine alfabetico, per toponimi o, come accennato precedentemente, eseguendo una ricerca più mirata, chiedendo al motore di ricerca di estrarre solo i documenti il cui campo *topography* contenesse il toponimo indicato nell'apposito form.

- Il campo *subject* avrebbe invece riassunto i topics salienti di ciascun testo, anch'esso reso come criterio d'ordinamento di tutti i documenti o come target di una ricerca per campo.

- Il campo *abstract* avrebbe fornito una sommarizzazione dettagliata dei contenuti del teso archiviato, costituendo quindi uno strumento di utilità enorme per quanti desiderassero consultare i testi estratti dalle proprie interrogazioni. [55]

3.4 visualizzare tutti i record di una tabella:

La prima applicazione di base che avrebbe dovuto essere realizzata era una pagina che permettesse all'utente di visualizzare tutti i record contenuti in una tabella.

Per la tabella *libri* doveva essere scritta una procedura in *Php* che andasse a scorrere tutti i record del database *bibliografiapiste* e estraesse il contenuto di ciascun record secondo criteri prestabiliti e/o scelti dall'utente[56] ed infine generasse una o più pagine *Html* contenente/i i dati estratti da visualizzare sul Browser.

La procedura sarebbe stata costituita quindi da due parti principali:

---

[54] E cioè cercando tutti i documenti, ad esempio, scritti da un certo autore, pubblicati in un certo anno e magari con soggetto un particolare argomento.

[55] Vi è inoltre un campo chiamato *Free*.
Esso si è reso necessario durante lo sviluppo dei sistemi di I.R. interni al portale, più precisamente utilizzato per l'implementazione delle procedure di ricerca libera, ossia un'interrogazione non vincolata a determinati campi del testo, ma mirata al reperimento dei documenti tramite il loro contenuto.
La realizzazione della ricerca *libera per parole chiave* è descritta successivamente.

[56] ordinare i record di una tabella secondo un determinato campo, e disporli poi in ordine alfabetico crescente o decrescente, ecc…

- La prima, del tutto in *Php* avrebbe contenuto le istruzioni necessarie alla connessione con il database e quindi con la tabella giusta.
  Essa avrebbe poi ordinato tutti i record della tabella selezionata a seconda del criterio (stabilito dall'utente) ricevuto come variabile dal form di ordinamento, e quindi avrebbe estratto gli elementi ordinati caricandoli in una variabile locale.

- La seconda parte della procedura, in *Html*, avrebbe gestito la formattazione grafica dell'interfaccia, la grafica della pagina ed i menù di dialogo con l'utente.
  Al suo interno vi sarebbe essa avrebbe richiamato la parte in Php come una funzione ritorsiva, e generando quindi i contenuti finali in base ai dati trasmessi dal richiamo degli script Php.

La procedura in *Php* avrebbe dovuto aprire una sessione di dialogo con il *Database*.[57]

Durante questa sessione di dialogo la procedura avrebbe dovuto, tramite le coordinate giuste, connettersi alla tabella desiderata, scorrerla tutta ed estrarre i propri record salvando il proprio contenuto in una variabile locale.

Prima però di questo processo di salvataggio sarebbe stato necessario che, se vi fosse stato un criterio di ordinamento scelto dall'utente, la procedura

[57] Il dialogo tra Php e MySql è eseguibile tramite funzioni, ed inoltre è possibile inserire degli script capaci di controllare la buona riuscita dell'apertura di sessione, e di comunicare in caso di problemi la chiamata al database non eseguita. Per una sintassi sia in Php che in Sql è consigliabile consultare:
Php: http://www.php.net/manual/ con un elenco completo e dettagliato di funzioni, script e la loro sintassi
MySql: http://www.mysql.com/ , e http://www.thickbook.com/extra/php_datatypes.phtml per una lista completa dei tipi di dati supportati dal database , con relative dimensioni e restrizioni applicabili.
Le funzioni di Php che consentono la comunicazione con un Server di Database MySql sono comunque numerose, nel caso specifico è comunque opportuno limitarsi a citare le seguenti: *mysql_connect*("nomeserver", "nomeutente", "passwutente"), funzione che stabilisce una connessione con un Server di Database MySql, la cui sintassi richiede le coordinate del server, e quindi il suo nome o Host, ed i dati utente ossia Login e Password perché il Server conceda l'accesso o meno e stabilisca per la sessione avviata i permessi attribuiti al determinato utente. Per tutelare una possibile mancata connessione è possibile apporvi la funzione *die*("messaggio") che offre una vera e propria via di scampo in caso di situazione critica (come la funzione *exit*). In questo caso, se non riuscisse la connessione con il database, è possibile utilizzare questa riga di codice per visualizzare un messaggio personalizzato e, contemporaneamente fermare lo script di dialogo per evitare un possibile loop del sistema (un ciclo infinito). Nel caso specifico le due funzioni saranno accoppiate in questo modo: *mysql_connect*("nomeserver", "nomeutente", "passwutente") *or die*("impossibile stabilire una connessione").

ordinasse i record estratti secondo il criterio predefinito e poi salvasse i dati ordinati all'interno della variabile locale.

Gli *script* che avrebbero gestito il passaggio dei dati dalla tabella del database alla variabile locale non avrebbero eseguito il processo direttamente, cioè non avrebbero letto e copiato i dati contenuti nel database, ma avrebbero utilizzato degli array di appoggio: essi sarebbero stati tanti quanti i campi della tabella di cui avessimo desiderio di visualizzarne il contenuto,[58] ed avrebbero dovuto avere tanti elementi quanti i record contenuti all'interno della tabella.

Gli *script* di *lettura*(*database*) **à** *scrittura* (*array*) sarebbero stati inseriti all'interno di un ciclo, che gli avrebbero resi ricorsivi, e cioè gli avrebbero ripetuti sino a quando la tabella del database non fosse stata del tutto scorsa.

Ogni array avrebbe quindi contenuto al suo interno il valore del campo di una determinata riga, e quindi tutti gli *array* di una sessione di ciclo avrebbero rappresentato una riga (cioè un elemento) della tabella, e dunque alla conclusione di ogni ciclo sarebbe stato necessario salvare tutti i dati contenuti in ciascun array all'interno di una variabile.

La variabile, quindi, alla fine di ogni iterazione, avrebbe contenuto tutti i dati relativi ad un elemento della tabella, e quindi applicando la procedura alla tabella *libri* la variabile avrebbe contenuto la rappresentazione di un testo.

L'ordine con cui gli array sarebbero stati caricati all'interno della variabile avrebbe costituito l'ordine con cui, alla conclusione della procedura, sarebbero state visualizzate le diverse voci di ciascuna scheda bibliografica.

Ad esempio, se avessimo salvato nella variabile il valore dell'array *autore* e poi quello dell'array *titolo*, oppure ne avessimo invertito l'ordine, non avremmo ottenuto la stessa visualizzazione della scheda bibliografica.[59]

Il processo di salvataggio dei dati all'interno della variabile, avrebbe sopperito non solo al trasporto dei dati dal database verso la sua visualizzazione finale, ma anche all'ordinamento delle voci della scheda bibliografica (o di un toponimo o di un testo geroglifico); inoltre sarebbe stato necessario inserire adesso tutti i Tags Html necessari alla formattazione grafica del testo contenuto, soluzione non solo estetica, ma necessaria per una chiara comprensione del contenuto poi visualizzato.

Php riesce a rendere tutto questo processo di scripting del tutto invisibile all'utente, per il quale tutta questa parte di procedura, anche visualizzando il codice della pagina apparsa in *layout* sul proprio *browser*, sembrerà non esistere.

A questo punto, definite le tappe principali della procedura di visualizzazione di tutti i record di una tabella, sarebbe stato opportuna implementarla finalizzandola al caso specifico: la tabella *libri*.

3.4.3 visualizzazione dei documenti della bibliografia

[58] e quindi in questo caso tutti i campi delle tre tabelle escluso il campo *id*, che occorre solo per la gestione da parte dell'amministratore

[59] ricordiamo che ogni array contiene il valore di un campo di un record, e cioè una voce della scheda bibliografica di un testo archiviato, o di un toponimo o di un testo geroglifico

La tabella libri, progettata precedentemente, sarebbe stata costituita da alcuni campi, dei quali quasi tutti avrebbero potuto essere considerati come possibili criteri per la costruzione di un indice utile alla consultazione dei documenti archiviati.

Come da progetto, sarebbe stato opportuno poter visualizzare ed ordinare tutti i documenti della bibliografia lasciando all'utente la scelta del campo che più ritenesse utile, e quindi, assunto quanto eseguito per la costruzione dell'indice dei testi geroglifici, sarebbe stato necessario realizzare tante pagine quanti fossero stati i campi adottabili come criterio d'ordinamento.

Questi ultimi consistevano in:

- *autore*,
- *titolo*
- *Isbn o Issn*
- *tipo di documenti*
- *soggetto*
- *toponimi*
- *data di pubblicazione*

Sarebbe stato necessario costruire un menù iniziale dal quale l'utente avrebbe potuto scegliere il criterio di ordinamento dei record contenuti nella tabella *libri*.

Per ogni criterio vi sarebbe stata una pagina in *Php* che avrebbe generato la lista dei documenti orinandoli a seconda del campo selezionato dall'utente, e quindi vi sarebbe stata la pagina *aut.php* per un indice dei testi per autore, la pagina *tit.php* per ordinare i documenti per titolo, ecc…

Ognuna di queste pagine avrebbe dovuto connettersi alla tabella *libri* del database *bibliografiapiste*, ed estrarre tutti i *record* ivi archiviati, disponendoli secondo l'ordine prestabilito, e visualizzando tutti i dati della scheda bibliografia ad eccezione del campo *abstract*, che invece sarebbe stato rappresentato all'interno della pagina *doc.php* puntata da un link ipertestuale dinamico per ciascun record.

In pratica il campo *abstract*, avrebbe dovuto contenere un link dinamico che, in dipesa dal documento (elemento della tabella) al quale esso apparteneva, avrebbe puntato alla pagina *doc.php* passandole tramite variabile un identificativo (il campo id della tabella libri) per far in modo che *doc.php*

selezionasse e dunque visualizzasse i valori archiviati di quel determinato elemento dalla tabella libri.

Potremmo dunque vedere le singole procedure di ordinamento dei testi, a parte per l'estrazione e la visualizzazione dei dati contenuti nella tabella, peculiari per la costruzione di un indice puntato e dinamico che avrebbe permesso una prima consultazione bibliografica più efficiente e meno appesantita dalle parti di testo costituenti i vari abstract dei testi rappresentati, anche perché sarebbe stato molto probabile che un utente, alla prima interrogazione dell'archivio, pur scegliendo l'ordinamento di tutto l'archivio, magari avrebbe desiderato consultarne il riassunto solo di alcuni.

Il sistema così creato avrebbe consentito la rappresentazione di tutti i documenti tramite una sola coppia di pagine dinamiche: la prima, dipendente dall'opzione di ordinamento preferita dall'utente (la chiameremo "scelta.php" che avrebbe potuto essere aut.php, tit.php, ecc…) *scelta.php*, avrebbe fornito la lista degli elementi, e la seconda *doc.php* avrebbe invece descritto l'elemento selezionato, sempre la stessa pagina con lo stesso codice, ma dai contenuti diversi a seconda dell'identificativo trasmesso da *scelta.php*.

Il cuore di questa struttura sarebbe stato dunque il processo di costruzione della lista dei testi: la variabile che avrebbe accolto il valore di *abstract* avrebbe dovuto ricevere il codice *Html* relativo alla creazione del link verso la pagina *doc.php*, e tramite lo script *?id=$id* specificando a quale record della tabella *libri* la pagina *doc.php* avrebbe dovuto puntare.[60]

Ogni documento avrebbe dunque puntato alla pagina, *doc.php*, i cui contenuti sarebbero stati generati dinamicamente a seconda del campo id del documento stesso.

---

[60] In pratica la pagina *doc.php* può essere considerata come una sorta di maschera con delle fessure capaci di far trasparire tutti i dati di un solo documento. La selezione dell'utente di un testo dalla pagina *scelta.php* può invece essere vista come una mano che posiziona questa maschera, la cui posa viene dettata dal link ipertestuale nel codice *?id=$id* in *scelta.php*.
Il codice del link ipertestuale è (nel caso in cui l'utente avesse deciso un ordinamento per titolo):
$abstract = " Visualizza una sommarizzazione del documento";
dove $ abstract è la variabile finale che accoglierà il link ipertestuale dinamico,
$id è l'array che contiene il valore del campo id della tabella, ossia l'identificativo che permetterà la selezione di un determinato elemento,
il tag Html Visualizza una sommarizzazione del documento assegna un link ipertestuale che punta alla pagina doc.php riferito al testo (in questo caso) o più correttamente a ciò che precede il tag con cui si interrompe il link ipertestuale, e quindi nel caso specifico, al valore contenuto nella variabile $id, mentre il codice ? assegna (operatore =) ad una variabile locale, id, il valore contenuto nell'array $id, passandola come parametro alla pagina *doc.php* che si riferirà dunque all'elemento della tabella libri con campo id = al valore della variabile $id.
Quindi ipotizzando che il valore di $id fosse stato *3* link ipertestuale attribuito al campo abstract risulterebbe : Visualizza una sommarizzazione del documento.

Questa pagina avrebbe quindi dovuto essere costituita da una procedura capace di connessione alla tabella *libri* del database *bliografiapiste*, e tramite lo stesso ciclo di trasferimento dati *database* **à** *array di appoggio* **à** *variabile di visualizzazione finale*, avrebbe dovuto essere capace di estrarre il contenuto della scheda di ciascun documento e di generare il relativo link *doc.php*.

La pagina *doc.php* avrebbe dunque fornito la scheda completa del documento selezionato dall'utente, completa questa volta del campo abstract, ottimo strumento per la comprensione generale delle tematiche trattate.

Tramite questa struttura dinamica l'utente avrebbe la possibilità di visualizzare tutti i documenti della bibliografia praticamente secondo ogni campo della tabella *libri*, ciascuno dei quali utilizzabile come criterio di ordinamento degli stessi elementi.

3.4.4 ricerca per parola chiave

Le strutture precedentemente implementate avrebbero permesso una consultazione della bibliografia già abbastanza personalizzabile da parte dell'utente, tuttavia essa era vincolata al fatto che la visualizzazione avrebbe riguardato tutta la bibliografia digitale, e non vi sarebbe stata la possibilità di poter interrogare l'archivio con una *query*,[61] ed estrarre quindi solo gli elementi che corrispondenti ad un'interrogazione più precisa.

La struttura che avrebbe permesso l'interrogazione all'archivio avrebbe costituito la risorsa di *I.R.* principale della biblioteca digitale, e come da progetto avrebbe dovuto articolarsi in due tipi di ricerca per parola chiave:[62]

1. *ricerca per campi*: dove la *query* sarebbe stata digitata dall'utente tramite un form che avrebbe presentato diversi campi sotto cui vincolare e ricercare la stringa inviata,

2. *ricerca libera*: la *query* sarebbe stata svincolata dai campi rappresentati dal form precedente, ed avrebbe invece interrogato la tabella libri nel campo free, costruito per rappresentare i contenuti ed i temi principali del record archiviato.[63]

---

[61] Con il termine *query* si intende un'interrogazione, cioè è l'insieme delle parole, o la parola, con cui si interroga, tramite un motore di ricerca, una banca dati per estrarne un elenco relativo all'argomento richiesto.

[62] La *parola chiave* (*keyword*) è la parola che definisce la richiesta effettuata ad un motore di ricerca e generalmente indica anche l'ambito di appartenenza a cui dovrebbero appartenere le pagine web trovate.

[63] Per contenuto ed argomenti principali del record non è inteso quanto archiviato nel campo subject e tanto meno nel campo abstract. Il campo Free, la cui descrizione averrà di seguito, è dichiarata come tipo di dato longtext, ed accoglierà delle chiavi identificative che faranno

Un *menù* avrebbe permesso all'utente di scegliere se effettuare una ricerca libera, svincolata dai campi bibliografici, oppure se vincolare la propria interrogazione ai campi della scheda bibliografica. Il menù avrebbe consentito di raggiungere gli appositi *forms*, costituenti anch'essi da realizzare in *Php*.

Ogni *form*, infatti, avrebbe dovuto contenere il codice di programmazione necessario a ricevere la *query* digitata, le opzioni di interrogazione, e quindi andare ad effettuare la ricerca interrogando la tabella *libri* del database *bibliografiapiste*.

Poi avrebbe dovuto contenere tutte le funzioni necessarie all'estrazione dei record attinenti alle *keywords* immesse dall'utente, e dunque avrebbe dovuto generare un *indice puntato* di documenti ordinandoli secondo i criteri prestabiliti.[64]

Ogni elemento dell'elenco puntato, rappresentazione di un documento, così come quanto realizzato per *l'ordinamento per campi* della bibliografia, avrebbe puntato a *doc.php*, la pagina che avrebbe descritto visualizzando anche il campo *abstract*, la scheda bibliografica completa dell'elemento selezionato.

La prima pagina, *key.php*, di facile realizzazione, avrebbe contenuto una succinta introduzione al funzionamento ed alle prestazioni del motore di ricerca che ci si accingeva ad usare.

3.4.6 ricerca vincolata ai campi

La *ricerca per campi* avrebbe dovuto sopperire alla possibilità di effettuare un'interrogazione al database tramite *query* da compilare nell'apposito form, la cui stringa (una o più parole) avrebbe dovuto essere ricercata all'interno del campo corrispondente a quello del form in cui l'utente lo avesse digitato.

In un archivio, come nella bibliografia digitale in questione e come già molto abbondantemente descritto precedentemente, ogni documento sarebbe stato rappresentato da da più campi organizzati in una tabella, ma non tutti i campi sarebbero stati necessari per la ricerca ed il reperimento dei documenti in modo efficiente: solo i più rappresentativi sarebbero risultati importanti ai fini delle operazioni di *I.R.* da eseguire sul corpus di testi.

Questi sarebbero stati come in qualsiasi sistema di biblioteca digitale:

- *titolo*,
- *autore*,

---

riferimento ad un'ontologia rappresentata anch'essa in una tabella del database bibliografiapiste

[64] Per indice puntato si intende quella struttura di link già incontrata in *scelta.php*
In questo caso i links sarebbero stati interni al sito stesso, ma successivamente vedremo che sarà possibile realizzare un indice puntato di link esterni.

- *soggetto*, nel caso in questione, un corpus di testi egittologici, il soggetto sarebbe comunque appartenuto a questi particolari campi semantici

- *topografia*, campo ponderante per l'estrazione di determinati documenti rispetto ad altri

Il *form* destinato a ricevere la *query* digitata dall'utente avrebbe dunque dovuto contenere 4 moduli entro cui poter scrivere una stringa di ricerca: le 4 stringhe sarebbero state caricate su una variabile locale e trasmesse alla procedura di ricerca.

La prima funzione da implementare per il funzionamento corretto della procedura di ricerca sarebbe consistita nel testare che almeno uno dei quattro moduli del form fosse stato riempito, condizione essenziale prima di effettuare un'interrogazione inutile verso il database che avrebbe avuto come risultato un errore da parte di *MySql* stesso.[65]

Il caso più semplice da rendere funzionante avrebbe coinciso con un solo campo riempito, e dunque la procedura di ricerca avrebbe dovuto gestire l'interrogazione esclusivamente verso il campo selezionato.

Era necessario inserire il codice opportuno per far estrarre dal database con un ciclo contenente le stesse istruzioni (passaggio di dati da campi di ciascuna riga della tabella libri ad array di appoggio e dunque alla variabile di visualizzazione finale) usate per l'estrazione dei dati nelle precedenti interrogazioni, ma avrebbe dovuto essere posta una condizione vincolante l'estrazione: il campo della tabella e a quello corrispondente selezionato nel

[65] Il test da verificare avrebbe utilizzato una classica struttura di controllo *if – else*. Il controllo sarebbe stato inserito prima di effettuare la chiamata ricorsiva all'interrogazione verso il database (la funzione *Select*) ed avrebbe dovuto coprire due casi:

- il caso in cui l'utente non avesse inserito alcun carattere all'interno di nemmeno un *campo* del *form*, implementato nell'istruzione *if (($keyN != "")* dove la variabile *$keyN* con *1<=N<=4* rappresenta le variabili locali dichiarate per accogliere le stringhe immesse dall'utente, ed il codice *“”* corrisponde all'assenza di carattere e *!=* è l'operatore logico indicante la condizione di non verifica (leggi se il valore di keyN è diverso da “”),

- il caso in cui l'utente, accidentalmente o volente, avesse inserito almeno in un campo il *carattere di spazio* (*_blank*) ed avesse confermato la ricerca di quando digitato, implementata dal codice *if (($keyN != ‘ ‘)* dove ‘ ‘ indica il carattere di spazio.

La procedura, dunque avrebbe contenuto entrambe le condizioni prima della chiamata ricorsiva verso se stessa.

form avrebbero dovuto contenere lo stesso valore;[66] la condizione doveva essere espressa durante la funzione *Select*.[67]

Tuttavia questa struttura era limitata all'estrazione di determinati elementi dalla tabella libri rispetto ad una *query* vincolata ad un solo campo, mentre l'obbiettivo sarebbe stato quello di poter rendere la ricerca il più precisa possibile, specificando gli altri campi.

Nel caso in cui, cioè, l'utente avesse riempito più di un modulo del form, il motore di ricerca avrebbe dovuto estrarre dalla tabella libri solo quei record che avessero soddisfatto contemporaneamente la struttura di confronto elaborata precedentemente riguardo ad un solo campo.[68]

Ad esempio: ipotizzando che l'utente avesse digitato nel 1° campo la stringa x, lasciato in bianco il campo 2° e 4°, mentre nel 3° egli avesse inserito la stringa y, la procedura avrebbe dovuto estrarre e visualizzare solo quei record della tabella libri che avessero contenuto nel valore dei propri campi corrispondenti al 1° e 3° del form, rispettivamente, sia la stringa x sia la stringa y.

Non avrebbe avuto molto senso abilitare il motore di ricerca ad estrarre quei documenti che avessero presentato l'una o l'altra stringa all'interno dei campi corrispondenti, poiché il risultato dell'interrogazione sarebbe stato equivalente a 2 interrogazioni distinte, ciascuna per campo, che avrebbe reso molto più leggero il processo di estrazione dei documenti.

Riprendendo quindi il codice utilizzato per l'estrazione dei documenti relativi alla *query* vincolata ad un solo campo, l'operatore where della funzione select avrebbe dovuto contenere la casistica di contemporaneità di *nomecampotabella LIKE '%$keyN%'* per tutti i campi adottati come dominio di ricerca.[69]

---

[66] Trattasi di *Stringhe*, e quindi lo stesso valore consiste nella stessa successione di caratteri

[67] La condizione da verificare per l'estrazione del determinato record dalla tabella sarebbe stata eseguita dall'espressione where apposta alla funzione di select. La sintassi corretta doveva essere:
select [nomi campi] from [nome tabella] where [espressione]
dove nel caso specifico l'espressione di controllo avrebbe dovuto rappresentare il caso in cui il valore della variabile locale corrispondente al campo N fosse contenuto nel valore del campo corrispondente della riga selezionata della tabella libri, tradotto nella sintassi corretta:
where nomecampotabella LIKE '%$keyN%'
dove il codice *LIKE* indica il caso di coincidenza (x = y), mentre la sintassi *%variabile%* indica che il valore della variabile *$keyN* non doveva essere esattamente identificato all'interno del campo della riga della tabella, ma per rendere vera questa condizione esso avrebbe potuto anche essere contenuto al suo interno (ad esempio la stringa *'zian'*, con questa struttura di controllo, contenuta nel valore *'egiziano'*, avrebbe restituito un valore positivo al controllo, e quindi la funzione di *select* avrebbe avuto come vera la condizione per la quale effettuare l'estrazione del record.

[68] Vedi nota 46

[69] La sintassi della funzione di select sarebbe dunque dovuta essere:
SELECT nomi campi FROM libri WHERE author LIKE '%$key1%' AND title LIKE '%$key2%' AND subject LIKE '%$key3%' AND topography LIKE '%$key4%';

Dunque aver implementato il controllo per l'esecuzione o meno della funzione di *select* verso la tabella illustrato precedentemente[70] si sarebbe rivelato fondamentale anche in questo caso, perché la condizione vera e contemporanea di tutti i controlli si sarebbe verificata anche quando venisse digitato per sbaglio o per volontà (oppure al primo collegamento con la procedura *key.php* dove alle variabili *$keyN* il valore assegnato per default sarebbe stato nullo) il pulsante di *submit*[71] senza aver riempito alcun campo del form di ricerca.

La funzione dunque avrebbe estratto i record relativi alla *query* incrociata ed avrebbe generato una pagina dinamica contenente, come da progetto, l'indice puntato relativo ai documenti estratti, che avrebbe fatto riferimento alla pagina dinamica doc.php che, come già visto per l'ordinamento dei documenti per campo, avrebbe fornito la scheda bibliografica completa del documento selezionato.

3.4.7 ricerca libera

La procedura che avrebbe dovuto eseguire la ricerca libera per *keywords*, ultima opzione di ricerca che sarebbe stata fornita all'utente, avrebbe rappresentato l'applicazione più difficile perché ricca di punti da sviluppare per una sua corretta implementazione.

La sua realizzazione, infatti, non si sarebbe limitata ad una struttura di procedure e di pagine, ma avrebbe modificato profondamente anche l'architettura del database *bibliografiapiste* ed il sistema stesso con cui finora erano state progettate ed implementate le sessioni d'interrogazione e di estrazione dei dati dalle tabelle del database.

La procedura, infatti, avrebbe dovuto eseguire un'interrogazione verso la bibliografia digitale che non fosse però limitata ai campi specifici della rappresentazione della scheda bibliografica.

Prima di tutto, essendo la *query* svincolata da ogni sottodominio,[72] sarebbe stato necessario fornire la possibilità di distinguere un'opzione relativa alla struttura fisica della *query* stessa.[73]

Il codice, sulla base di quanto spiegato precedentemente è chiaro: la funzione di select avrebbe estratto i record della tabella libri solo se questi rispettavano la contemporanea condizione di *TRUE* delle 4 strutture di controllo. L'esecuzione della ricerca con questo tipo di vincolo su più domini viene detta *ricerca incrociata*.

[70] Vedi nota 44

[71] Il pulsante di submit una volta clickato, nel caso specifico, da inizio alla procedura in php, ossia all'interrogazione verso il database

[72] Mentre nelle applicazioni precedenti i *campi* del *form* *key.php* verso i quali era destinata l'interrogazione alla tabella libri, avrebbero costituito una specifica di precisione fondamentale per una ricerca per *keywords* di una sessione di *I.R.* precisa, finalizzata all'estrazione di certi ben precisi documenti, per questa applicazione il punto di vista cambia, ed il vincolo verso i campi della bibliografia digitale è visibile come una limitazione a dei sottodomini che avrebbero compromesso l'estendibilità della ricerca a *tutto* il *corpus della bibliografia* stessa.

Se essa fosse stata costituita da una sola parola, [74] il problema non sarebbe sussistito, poiché la parola sarebbe stata ricercata tramite le strutture che saranno illustrate successivamente, e l'interrogazione avrebbe provveduto all'estrazione dalla tabella *libri* dei testi corrispondenti.

Nell'altro caso[75] sarebbe stato necessario distinguere le seguenti opzioni:

- *ricerca di tutte le parole*: dove cioè si andasse a cercare quei documenti che avessero contenuto nelle strutture apposite tutte le parole presenti nella stringa di *query*, anche in ordine sparso e non necessariamente seguendo la sequenza digitata dall'utente. Le parole della *query* sarebbero state dunque inserite all'interno di un'espressione costituente la struttura della funzione di *select*, le cui singole strutture di controllo, ciascuna per parola, sarebbero state connesse da una relazione di tipo *AND*, (vedi a fine capitolo la figura relativa)

- *ricerca di almeno una delle parole* : la procedura avrebbe dovuto estrarre tutti i record della tabella nel cui campo dedicato alla ricerca libera fossero state contenute almeno una delle parole costituenti la *query*. L'espressione della funzione di select avrebbe dunque connesso le singole strutture di controllo tramite una relazione di tipo *OR*,[76] (vedi a fine capitolo la figura relativa)

- *ricerca della frase esatta*: la procedura avrebbe dovuto considerare la *query* come una stringa intera, senza suddividerla in parole, ed avrebbe dovuto estrarre i documenti della tabella libri che avessero contenuto

---

[73] La query può essere fisicamente costituita da una o più parole, ma nel caso specifico, così come nella maggioranza delle query ipotizzabili, sarà di seguito sviluppato il caso che la query fosse stata composta da più di una parola

[74] Ipotizzando il caso in cui la *query* fosse composta da *N* parole, la funzione di select avrebbe cercato nella tabella apposita alla *ricerca libera* le *N* parole e cioè:
SELECT [nomi campi] FROM [nome tabella] WHERE nomecampo LIKE '%parola1%' AND nomecampo LIKE '%parola2%'… AND nomecampo LIKE '%parolaN%';
La sintassi corretta non lascia equivoci: sarebbe stato estratto solo quel record che avesse contenuto nel campo nomecampo tutte le parole della *query* a prescindere dal loro ordine di successione all'interno del campo.

[75] Cioè quando la Query sarebbe stata composta da più parole, il più frequentemente incontrato

[76] La sintassi della funzione di select sarebbe risultata identica alla precedente, tranne che per la struttura di controllo:
SELECT [nomi campi] FROM [nome tabella] WHERE nomecampo LIKE '%parola1%' OR nomecampo LIKE '%parola2%'…OR nomecampo LIKE '%parolaN%';

esattamente la stringa di *query* al proprio interno.[77] (vedi a fine capitolo la figura relativa)

La scelta di una delle tre opzioni sarebbe dovuta avvenire all'interno dello stesso form che avrebbe accolto la *query* stessa, e sarebbe stato fornito tramite un menù a tendina.

Lo sviluppo delle prime due opzioni avrebbe implicato una manipolazione della *query* digitata dall'utente, infatti questa sarebbe stata trattata non nella sua integrità, ma elemento per elemento (e quindi parola per parola).

La riuscita delle prime due opzioni, quindi, sarebbe dipesa dall'implementazione di una procedura da applicare preventivamente alla *query*, detta di *tokenizzazione*, capace di scomporre la stringa in un array di parole il cui numero di elementi sarebbe coinciso con il numero delle parole costituenti la stringa iniziale, e dove ogni elemento ne avrebbe contenuta una (*token*), e solo a questo punto sarebbe stato possibile sviluppare le opzioni di ricerca.

L'obbiettivo principale della ricerca libera, tuttavia, non sarebbe stato solo il servizio delle tre opzioni di ricerca applicabili alla *query*, ma avrebbe dovuto consistere nello svincolare la ricerca per parola chiave dai campi della tabella libri, per poter eseguire un'interrogazione ad un livello concettuale più astratto, più elevato, andando cioè ad interrogare circa il contenuto e gli argomenti principali dei testi della bibliografia digitale.

Infatti tutti i tre criteri di ricerca erano riferiti ad una struttura contenuta nel database bibliografiapiste ancora indefinita, che avrebbe dovuto archiviare la rappresentazione dei contenuti dei testi.

Questa rappresentazione avrebbe dovuto avere la caratteristica di rimanere invisibile all'utente, ma di riuscire a rendere reperibili tutti i documenti della *bibliografia* tramite delle *chiavi* comuni ad essi, e connotate in modo che rispecchiassero i criteri logici di una possibile interrogazione.

Le chiavi di ricerca avrebbero rappresentato gli argomenti principali di ciascun documento, e quindi avrebbero dovuto puntare a tutti gli elementi della bibliografia i cui contenuti fossero riconducibili al determinato argomento, mentre ogni documento avrebbe potuto puntare anche a tutti gli argomenti identificati all'interno della bibliografia digitale.

Sarebbe stato necessario costruire due nuove strutture all'interno del database bibliografiapiste:

- una nuova tabella che avrebbe accolto ed archiviato gli argomenti detta *infor*, indicizzati in modo da poterli facilmente relazionare,

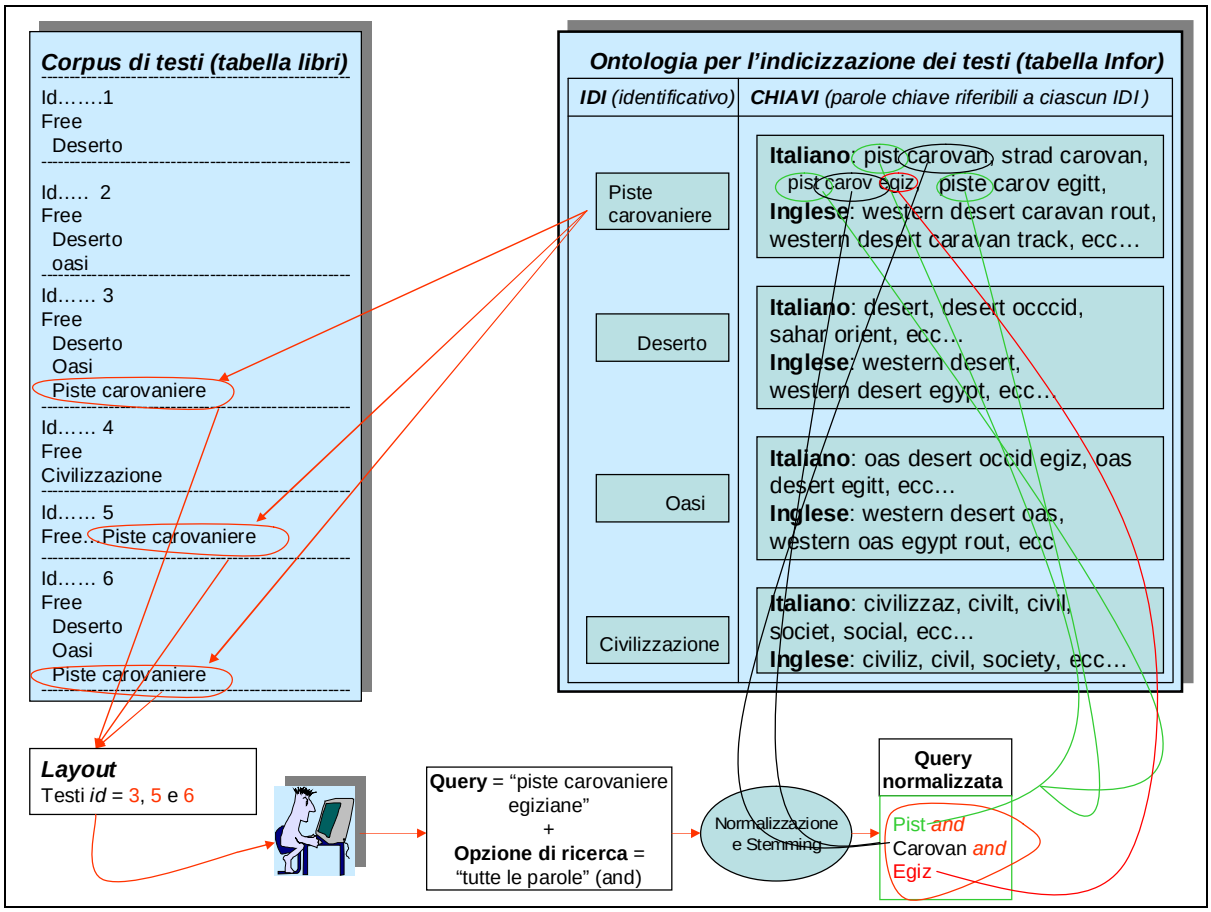
[77] Ipotizzando che la *query* fosse contenuta nella variabile *$query*, la funzione di *select* avrebbe avuto la seguente sintassi:
SELECT [nomi campi] FROM [nome tabella] WHERE nomecampo LIKE '%$query%';

- un nuovo campo all'interno della tabella libri che avrebbe invece contenuto la relazione agli argomenti di cui il testo avesse trattato, detto *free*, costituito dallo stesso indice che avrebbe connotato la tabella *infor*.

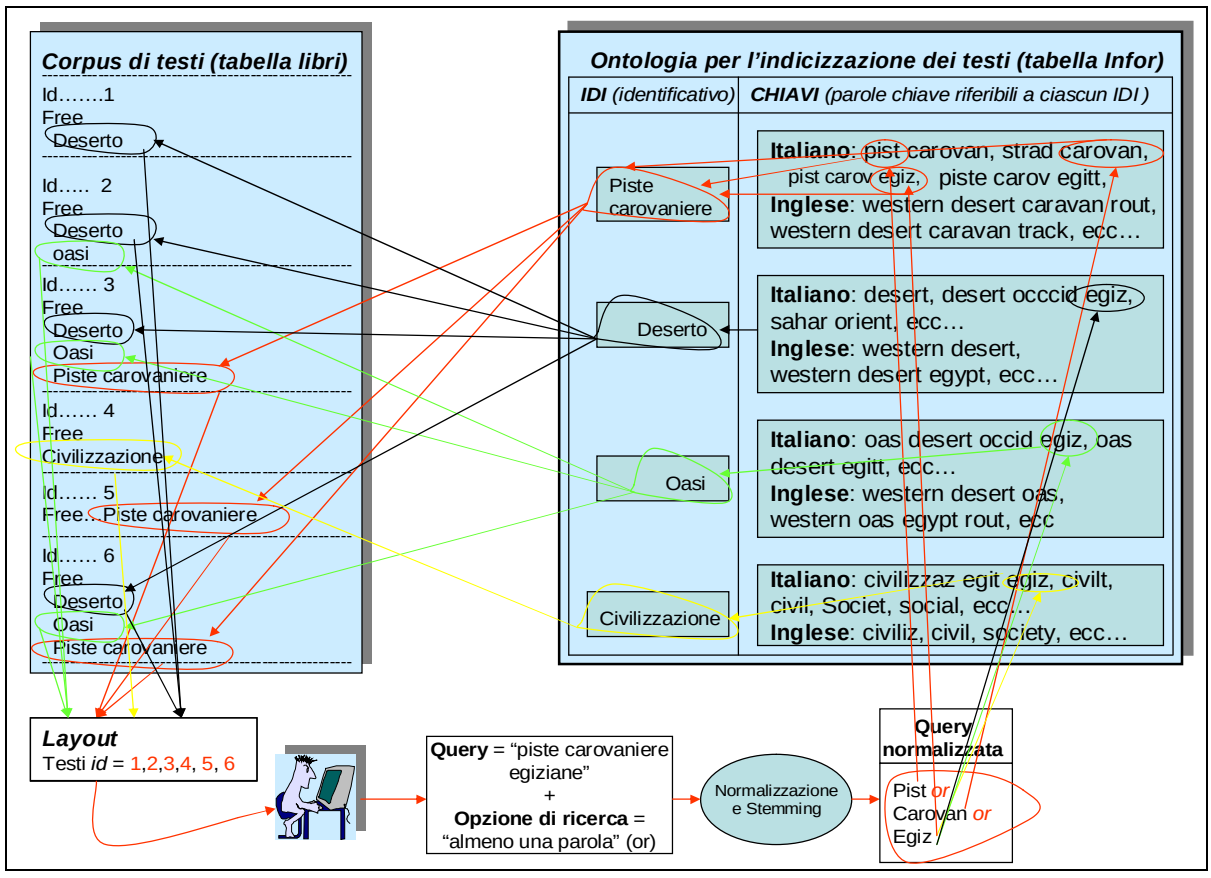
Il funzionamento della ricerca libera sarebbe stato diverso dalle interrogazioni precedenti: infatti la *query* sarebbe stata manipolata nel caso in cui si desiderasse effettuare la ricerca di tutte o di almeno una delle parole dell'interrogazione, e quindi l'interrogazione non avrebbe più riguardato direttamente la tabella libri, ma sarebbe stata destinata alla tabella infor, contenente una struttura capace di rappresentare i concetti generali contenuti all'interno della bibliografia, e dove presente, avrebbe estratto il valore del campo di indice di questi elementi.

Questi valori sarebbero poi andati a costituire una nuova espressione di una funzione di *select*, stavolta destinata ad interrogare il campo *free* della tabella *libri*, e la funzione avrebbe estratto solo i documenti nel caso in cui i due indici avessero coinciso, e cioè se il valore dell'espressione di *select* fosse corrisposta al valore del campo *free*.[78]

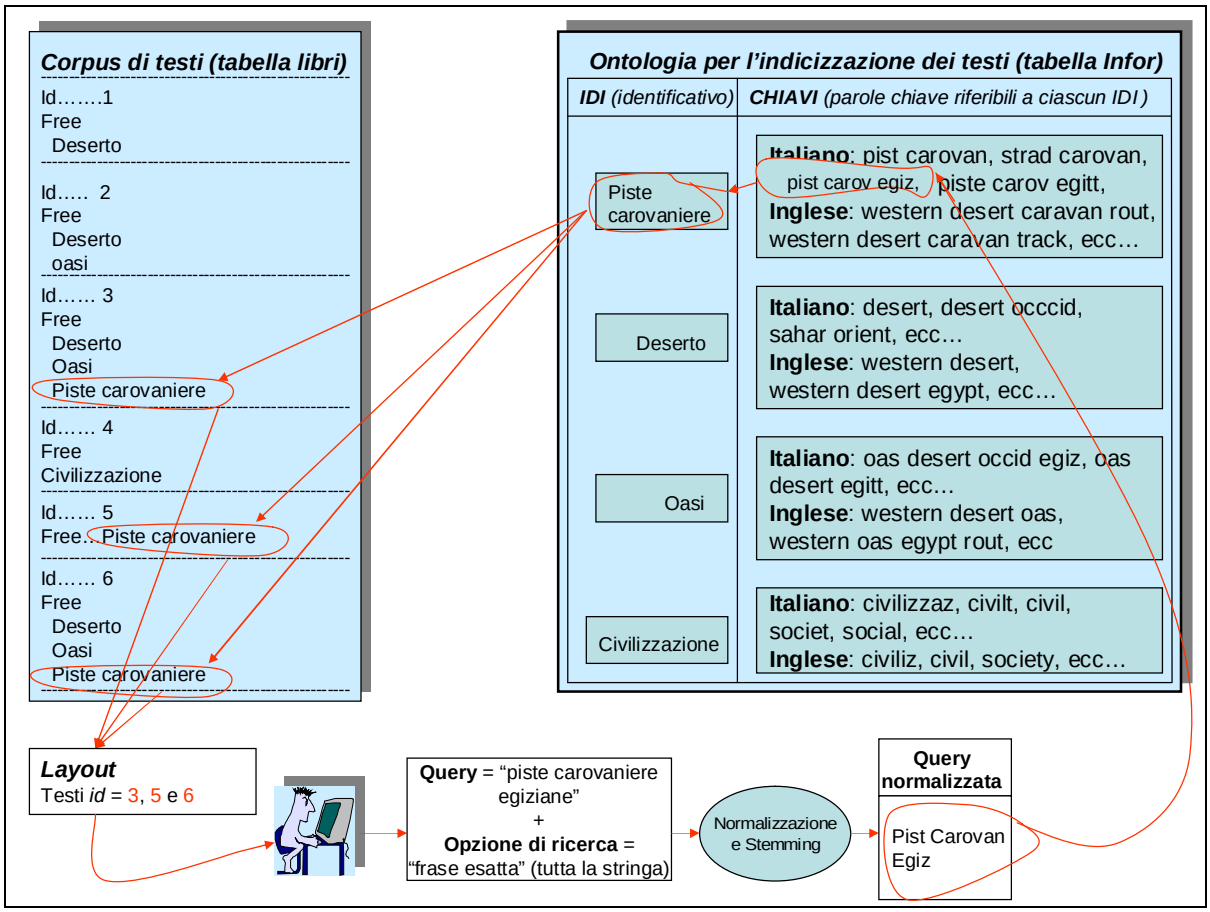
[78] La prima funzione di select sarebbe stata indirizzata alla tabella infor. Da qui avrebbe estratto il valore del campo che indicizza gli elementi (argomenti principali della bibliografia) che avrebbe caricato in una variabile.
La stessa variabile avrebbe poi costituito l'espressione di una nuova funzione di select destinata all'interrogazione del campo free della tabella libri contenente lo stesso sistema di indici usato per connotare la tabella infor.
Gli elementi estratti sarebbero stati quelli relazionati agli argomenti precedentemente identificati.



Interrogazione libera per parola chiave con opzione "tutte le parole"



Interrogazione libera per parola chiave con opzione "almeno una parola"



Interrogazione libera per parola chiave con opzione "frase esatta"

4 II livello di MLIA: applicazione e realizzazione di un CLIRs

4.1 la costruzione dell' ontologia infor

La tabella *infor* avrebbe dovuto raccogliere tutti gli argomenti principali contemplati all'interno della bibliografia digitale.
Essa sarebbe stata costruita in modo da costituire una *ontologia.*

L'*ontologia infor* avrebbe costituito lo strumento tramite cui realizzare la ricerca libera, perché questa fosse svincolata dai campi della tabella libri, e perché la ricerca dei documenti avvenisse ad un livello concettuale più elevato, astratto da ogni singolo elemento della bibliografia.

La sua struttura, in un certo senso, avrebbe dovuto ricordare quella di un insieme di campi semantici, dove gli elementi appartenenti ad ognuno di essi sarebbero state tutte quelle parole che lo evocassero, ed ogni campo semantico sarebbe stato identificato dal proprio nome. Implementando questa intuizione per la costruzione di una tabella all'interno del database essa sarebbe stata organizzata in soli 2 campi:[79]

idi :	argomenti della bibliografia
Chiavi :	parole appartenenti al campo semantico della metachiave

Questa tabella avrebbe costituito l'archivio degli argomenti principali contenuti all'interno della bibliografia, ad esempio:

idi :	Oasi
Chiavi :	Bahariya, oasi di Bahariya, oasi occidentali, oasi del deserto occidentale, L'oasi di Bahariya, nell'Oasi di Bahariya, Farafra, l'oasi di Farafra, oasi di Farafra, ecc…

Questa struttura avrebbe rappresentato dunque il concetto generale di *oasi*, nome registrato nel campo *idi*, mentre la serie di parole costituenti il campo chiavi avrebbero rappresentato le chiavi, appartenenti al campo semantico oasi.

La tabella *infor* sarebbe stata affiancata e relazionata alla tabella libri tramite un nuovo campo detto free, e contenente tutti i campi semantici di cui il documento avesse trattato, gli stessi del campo *free*.

[79] Momentaneamente è bene riferirsi all'ontologia infor nella sua versione monolingue. In realtà essa è una struttura multilingue, il cui numero di campi dipende dalle lingue rappresentate, e segue la regola seguente: *numerocampi = numerolingue rappresentate + 1*.

free :	argomento1, argomento2, … argomentoN

Tramite queste strutture il processo di interrogazione verso il database sarebbe risultato diverso da quanto precedentemente implementato: la *query* immessa dall'utente sarebbe stata utilizzata per l'interrogazione verso la tabella infor del database.

La funzione di *select* avrebbe estratto il valore del campo *idi* degli elementi della *ontologia infor* che avessero contenuto la *query*, o parti di essa, fra i valori del campo *chiavi*.[80]

Il codice estratto avrebbe costituito l'espressione della successiva funzione di *select*: essa avrebbe interrogato la tabella *libri* cercando ed estraendo solo quegli elementi che avessero presentato all'interno del campo *free*, valori uguali o contenenti le parti della *query* (corrispondenti ai valori del campo *idi* estratti precedentemente dall'*ontologia infor*) realizzando quindi un'interrogazione a livello concettuale e logico superiore perché eseguita tramite la tabella *infor*, contenente la rappresentazione astratta di un concetto principale tramite il suo nome nel campo idi, [81] e dei propri sinonimi, iponimi, ecc… archiviati nel campo chiavi.

4.2 query multilingue: realizzazione del C.L.I.R.

Le caratteristiche fondamentali della tabella infor, creata come un'ontologia per la rappresentazione dei contenuti semantici presenti nella bibliografia digitale, avrebbero dovuto essere:

- la propria indipendenza dagli elementi della bibliografia,
- la sua elasticità, ossia la possibilità di essere sempre ampliata di un nuovo campo idi nel caso venisse identificato un nuovo argomento da dover rappresentare

[80] La ricerca della *query* per intero sarebbe dipesa dall'opzione di ricerca della frase esatta selezionabile dall'utente. Altrimenti la *query* sarebbe stata considerata non più per intero, ma tokenizzata, e la funzione di select avrebbe organizzato i diversi tokens con relazioni di tipo *AND* o di tipo *OR* rispettivamente adottate in base all'opzione di ricerca di *tutte le parole* o di *almeno una parola*, sempre scelta dall'utente durante la compilazione del *form* di ricerca. Vedi anche

[81] Contemporaneamente anche codice identificativo, e quindi nominato campo *idi*.
Il fatto che esso sia rappresentato da un nome è semplicemente per abbreviare i tempi di identificazione del contenuto semantico del campo stesso.
In realtà il campo *idi*, così come il campo *free* della tabella *libri*, avrebbe potuto essere costituito da un indice qualsiasi, anche numerico.
La relazione fondamentale che avrebbe dovuto essere mantenuta, sarebbe stata l'equivalenza dei valori dei campi *idi* e *free* per relazionare ad ogni documento della bibliografia digitale gli argomenti contenuti.

all'interno della bibliografia, oppure la possibilità di aggiungere chiavi relative ad argomenti già esistenti.

L'indipendenza dalla tabella libri era realizzata fisicamente…questo perché la tabella infor è esterna ed amministrabile indipendentemente dall'archivio dei documenti della bibliografia digitale ai quali però era relazionata tramite i rispettivi campi *idi* **à** *free*.

L'elasticità della struttura sarebbe stata dovuta sia dal modo con il quale era stata concepita e realizzata, sia dalle buone prestazioni di *MySql* per la gestione di database.

Infatti esse sopperivano ad una carenza che altrimenti avrebbe compromesso l'implementazione di tutta l'architettura precedentemente illustrata: una volta costruito un campo, e dichiarati i tipi di dati che esso avrebbe contenuto, era necessario rispettare delle dimensioni massime dopo le quali all'interno del campo non sarebbe più stato possibile inserire nuovi valori.

Questo avrebbe significato che se un record della tabella infor avesse superato le dimensioni supportate per il campo chiavi, il record non sarebbe più stato aggiornabile, sempre ammesso che le dimensioni supportate avessero già sopperito all'inserimento di tutte le chiavi programmate.

L'architettura di *MySql*, invece, rappresenta la soluzione a questo problema strutturale.

Infatti sarebbe stato possibile, una volta raggiunte le dimensioni massime di un record, inserire un altro elemento all'interno della bibliografia, ed assegnarvi lo stesso valore (nome) nel campo idi, e quindi continuare nell'inserimento delle chiavi relative allo stesso argomento il cui sviluppo era stato bloccato da meri problemi di spazio.

Questa avrebbe consentito lo sviluppo sempre maggiore e sempre più preciso di chiavi (e combinazioni di parole) sempre più efficienti per il reperimento dei documenti tramite l'ontologia infor: teoricamente, per ogni argomento, avrebbero potuto essere costruiti infiniti record contenenti lo stesso valore nel campo idi, (nome dell'argomento) e chiavi diverse per la sua rappresentazione nel campo chiavi.

Questa ultima affermazione, "*chiavi diverse per la sua rappresentazione*", avrebbe costituito lo spunto per la realizzazione di una struttura di *I.R.* multilingue.

Infatti se avessimo considerato in modo del tutto oggettivo la rappresentazione all'interno di infor un argomento della bibliografia (e cioè dei record della tabella dove il campo idi fosse stato uguale), così come la rappresentazione di un lemma all'interno di un dizionario multilingue, avremmo potuto trovarvi questa relazione, perché entrambe le rappresentazioni erano costituite da:

- un *lemma* (*argomento* della *bibliografia digitale* nella tabella *infor*, voce del dizionario nel *vocabolario multilingue*),

- la sua *rappresentazione* (il campo *chiavi* in tutti gli elementi con *idi* identico di *infor*, le definizioni in ciascuna lingua della stessa parola nel *dizionario multilingue*)

e cioè entrambe le strutture erano realizzate attraverso un oggetto indicizzato (*voce del dizionario* ß à *argomento della bibliografia*) al quale erano riferite le proprie rappresentazioni (nel caso del dizionario multilingue le diverse definizioni, ciascuna per linguaggio, relative allo stesso lemma).

Queste deduzioni apparentemente banali, avrebbero permesso, previamente accertato che la tabella *infor* potesse accogliere elementi con lo stesso valore all'interno del campo *idi* ma con chiavi diverse attribuitegli, che la rappresentazione di ogni argomento avrebbe potuto essere indipendente dalla lingua nella quale fossero state espresse le *chiavi*.

Cioè che la tabella infor avrebbe potuto contenere uno o più record che avrebbe rappresentato quel determinato argomento con i campi chiavi espresse in una determinata lingua, e poi altri record rappresentanti lo stesso argomento (campo idi identico) con chiavi in un'altra lingua, e di nuovo altri elementi ancora che avrebbero rappresentato lo stesso argomento con chiavi in una terza lingua:

Nome campo	*Valore*	**Nome campo**	*Valore*
idi :	*Oasi*	**Chiavi** *(italiano)* **:**	*L'oasi di Bahariya, Oasi del deserto occidentale, ecc…*
idi :	*Oasi*	**Chiavi** *(inglese)* **:**	*Bahariya oasis, Egyptian western desert oasis, ecc...*
idi :	*Oasi*	**Chiavi** *(linguaX)* **:**	*chiave1X, chiave 2x, chiave3x… chiaveNX*

Questa struttura era applicabile teoricamente a qualsiasi lingua,[82] ed avrebbe consentito la realizzazione di quanto aspirato durante l'esposizione nella 2° parte relativa al concetto di *C. L. I. R.* [83]

[82] Nella tabella rappresentata dalla *linguaX*, le cui chiavi, *chiaveXN* rappresentano stringhe di caratteri così come le chiavi esemplificate per l'Italiano e per l'Inglese.
Teoricamente perché la sua realizzazione pratica avrebbe sicuramente incontrato nuove difficoltà per la sua corretta implementazione, difficoltà di livello strutturale dovute alla sintassi di ciascun linguaggio, oppure ai problemi derivati dall'uso di alfabeti diversi e quindi dalla difficoltà di reperire i fonte unico de per poter scrivere correttamente le singole parole, ecc…

Tramite questa struttura, la procedura di estrazione dei documenti avrebbe potuto estrarre gli stessi record dalla tabella libri a prescindere sia da quale lingua fosse stata usata per la redazione della scheda bibliografica, sia da quale lingua fosse stata utilizzata per comporre la *query* d'interrogazione.

Non solo, l'ontologia *infor*, divenuta multilingue, avrebbe consentito la possibilità per l'utente di formulare *query multilingue miste*, poiché l'interrogazione verso la tabella *infor* avrebbe avuto riferimento ai contenuti dei campi chiavi, ma avrebbe estratto lo stesso campo idi tramite cui interrogare la *bibliografia digitale*, ed il campo *idi* sarebbe stato raggiungibile sia tramite *parole chiave inglesi, italiane* o in *linguaX*.[84]

Schematizzando dunque i processi relativi ad una *query* di ricerca libera:

- L'utente avrebbe compilato il *form*, scelto l'opzione di ricerca,[85] e lanciata l'interrogazione,

- Il sistema avrebbe ricevuto la *query*, tokenizzata, ed eseguito la prima funzione di select estraendo per ogni tokens della *query* il campo idi corrispondente (se presente)

- Il sistema avrebbe generato una nuova funzione di select contenente i campi idi estratti dalla precedente interrogazione.
  Sarebbero così estratti i documenti attinenti all'espressione di select.

- I documenti estratti sarebbero stati visualizzati nel browser utente generando un indice puntato del tutto identico a quell'implementato per la realizzazione della ricerca per campi.
  Ogni documento sarebbe stato munito di link alla pagina *doc.php* che ne avrebbe visualizzato, se selezionato, la scheda bibliografica completa.

---

[83] Il concetto di *Cross Language Information Retrival, C.L.I.R.*. è stato ampiamente affrontato nella 2° parte a cui è consigliabile rifarsi per un'adeguata comprensione delle problematiche proposte.

[84] Il sistema implementato avrebbe estratto gli stessi documenti sia da una *query italiana*, ad esempio *le piste carovaniere*, sia da una *query inglese*,ad esempio *caravan's routes*, sia da una query mista, ad esempio *le piste of caravans*, grammaticalmente scorretta e praticamente improbabile, ma esempio che illustra efficacemente le potenzialità della struttura realizzata.

[85] L'opzione frase esatta, già sviluppata precedentemente, sarà ripresa successivamente. Per adesso è necessario esaminare le altre due opzioni, perché ad esse viene applicata la *tokenizzazione* della stringa di *query*.

Infor, la tabella inizialmente concepita per la semplice[86] realizzazione di una ricerca svincolata dai campi della tabella libri, e costruita per eseguire un'interrogazione al database ad un livello concettuale astratto e quindi superiore si era trasformata in una risorsa multilinguistica, realizzando efficientemente quanto proposto dall'aspetto del *C.L.I.R.* per un *Multi Languages Information Access* alla *bibliografia digitale*.

4.3 manipolazione della query per I.R.S. più efficiente

La struttura implementata per la realizzazione della ricerca libera per *keywords* si sarebbe basata dunque sull'elaborazione della stringa originale di *query*.

Essa sarebbe stata scomposta in *tokens*, i quali sarebbero stati cercati all'interno dell'*ontologia infor*, rappresentazione dei contenuti semantici dei documenti contenuti nella *bibliografia*, e dunque gli idi estratti dai documenti che avessero rispettato le condizioni della prima interrogazione, avrebbero costituito l'espressione della seconda *query* di *select* destinata all'interrogazione verso la tabella *libri*.

Tutti i processi di questa successione di operazioni sarebbero stati cruciali ed essenziali, e l'ottimizzazione anche di uno solo di essi avrebbe costituito un miglioramento di tutta l'operazione di *I.R.*.

La seconda parte del processo di interrogazione per *query libera* non sarebbe stato direttamente suscettibile di ulteriori miglioramenti, poiché l'estrazione dei documenti dalla bibliografia in modo efficiente sarebbe dipesa dal corretto funzionamento della prima funzione di *select*, ossia dalla capacità delle procedure implementate nel riuscire a risalire ai giusti campi *idi* della tabella *infor* dalle parole della *query tokenizzata*.

Dunque i miglioramenti al sistema avrebbero dovuto riguardare proprio quest'ultima parte citata, ed essere inserite nella successione delle procedure di interrogazione dopo il processo di *tokenizzazione* e prima della funzione di *select* delle parole *tokenizzate* della *query* verso l'*ontologia infor*.

Infatti la parte precedente risultava già ben architettata, e fino al momento in cui la *query* non venisse *tokenizzata* e quindi trasferita all'*array*,[87] non sarebbe stato possibile aggiungervi processi o modificarne le prestazioni.

Sarebbe invece stato possibile applicare tutta una serie di strutture, nell'ordine di quanto esaminato nella sezione dedicata all'*I.R.*, che avessero potuto eseguire delle operazioni di normalizzazione e di stemming sulla *query* inserita dall'utente.

In realtà un primo e flebile processo di *normalizzazione* della *stringa* di *query* sarebbe già avvenuta durante la sua *tokenizzazione*, ma vi erano altre

[86] L'attributo "semplice" è relativo a quanto sviluppato oltre il primo obbiettivo prefissato, e non vuole sminuire il notevole livello di efficienza che già avrebbe avuto un sistema dotato della semplice *ricerca libera monolingue*.
[87] *Array* che l'avrebbe contenuta parola per parola.

operazioni inseribili fra la *tokenizzazione* e la costruzione della prima funzione di *select*.

Un esame distaccato dal contesto in questione della stringa ridotta a tokens avrebbe potuto portare ad una visione astratta della stringa rappresentata all'interno dell'array ad un insieme di elementi,[88] ed ogni elemento avrebbe costituito un ciclo nel primo processo di select, visto che ognuna di queste parole avrebbe potuto essere una chiave potenziale riconducibile ad un argomento dell'ontologia *infor*.

Preso atto di quanto affermato, era evidente che la durata della prima interrogazione verso il database sarebbe stata in relazione diretta col numero di parole in essa cercate, [89] e dunque una prima ottimizzazione avrebbe potuto consistere nell'eliminazione di elementi superflui, nello specifico rappresentati da parole inutili e ridondanti quali gli articoli, le preposizioni semplici e composte, ecc…,il cui insieme è detto *stopwords*.

L'eliminazione delle *stopwords*, oltre a diminuire i tempi di esecuzione di tutta la prima funzione di select, avrebbe migliorato l'efficienza del funzionamento di tutta la struttura, poiché per le strutture di controllo usate all'interno delle procedure le parole di stop avrebbero costituito elementi che avrebbero portato a fraintendimenti: prendiamo ad esempio le due stringhe x = ' *il deserto occidentale* ' ed y = ' *deserto occidentale* '.

Ipotizziamo di eseguire due interrogazioni distinte, la prima con la stringa x e la seconda con la stringa y: secondo le strutture implementate fino a questo momento entrambe sarebbero state tokenizzate, e quindi avremmo ottenuto i seguenti array:

ArrayX

ArrayX[1]	*il*
ArrayX[2]	*deserto*
ArrayX[3]	*occidentale*

ArrayY

ArrayY[1]	*deserto*
ArrayY[2]	*occidentale*

Ogni elemento di ciascun *array* avrebbe poi costituito un'espressione di select verso la tabella *infor*, e quindi è subito evidente che mentre la funzione di select relativa alla *query* X avrebbe compiuto un ciclo in più della funzione di select relativa alla *query* Y peggiorando nella velocità di esecuzione.

Inoltre ipotizzando che l'utente avesse selezionato l'opzione di ricerca almeno una parola, allora i documenti estratti tramite la *query* X sarebbero stati gli stessi ottenuti con la *query* Y, ma se invece egli avesse preferito l'opzione

---

[88] I tokens, cioè le parole.

[89] Chiamando Tarray[n] il tempo impiegato nell'esecuzione della funzione di select dell'elemento N dell'array (token della stringa di query) è evidente che il tempo totale dell'esecuzione di tutta l'interrogazione sarebbe dato dalla sommatoria di tutti i Tarray[n] dove n = numero di parole della stringa di *query*, e dunque:

$$Ttotale = \sum (i = 1)(i = N)\, Tarray[i]$$

tutte le parole, i documenti estratti dalla prima funzione di select non avrebbero potuto essere gli stessi poiché l'articolo determinativo "il" costituiva una chiave da tenere in considerazione per il verificarsi delle condizioni necessarie all'interrogazione verso la tabella infor e dunque verso l'archivio dei documenti.

Se invece vi fossero state una serie di strutture capaci di riconoscere che l'articolo "*il*" avrebbe potuto essere eliminato, le *query* avrebbero potuto estrarre, come dovuto, gli stessi documenti.

La struttura da realizzare avrebbe dovuto quindi esaminare che ogni elemento dell'array non fosse stato una *stopwords*, e nel caso contrario provvedere alla generazione di un nuovo *array* privo dell'elemento da eliminare.

Ma vi sarebbero stati altri casi di ambiguità costituita semplicemente dal modo nel quale venivano scritti gli stessi concetti, e quindi essendo i concetti espressi gli stessi, se essi fossero stati stringa di una *query*, anche i documenti che alla fine l'interrogazione avrebbe dovuto estrarre dovevano essere gli stessi.

Riassumendo, prima di analizzare nel dettaglio l'implementazione delle procedure di *Tokenizzazione*, di *Stopwords* e di *Normalizzazione* e *Stemming*, il sistema di ricerca libera per *keywords* si sarebbe arricchito di queste due ulteriori strutture, che avrebbero permesso di ridurre i tempi di interrogazione verso il database, ed avrebbero eliminato degli elementi ridondanti, possibili fonti di ambiguità e di fraintendimento di *query* riconducibili allo stesso significato.

4.4 tokenizzazione

Per *tokenizzazione* si intende quel processo di manipolazione di una stringa attraverso il quale essa viene suddivisa in unità basilari (*parole*) dette *tokens*: ogni *token* è rintracciabile tramite un indice, e può essere connotato con delle informazioni aggiuntive, come la propria analisi sintattica e semantica.
Nel caso specifico la *tokenizzazione* avrebbe dovuto limitarsi ad una manipolazione fisica della stringa, e non sarebbe stato necessario connotare il *token* di alcuna informazione aggiuntiva, se non di un semplice indice numerico per la sua reperibilità.

La *tokenizzazione* sarebbe stata eseguita con poche righe di codice grazie alla vasta gamma di funzioni per la manipolazione delle variabili ed, in questo caso, più precisamente, delle stringhe di testo offerte da *Php*.[90]

---

[90] La struttura necessaria alla tokenizzazione della stinga di query sarebbe stata costituita dalla funzione explode. La funzione explode suddivide una stringa utilizzando un determinato carattere di separazione come criterio del troncamento delle parola della stringa, ed essa inserisce i valori ricavati in un array: la sua sintassi corretta sarebbe stata:
$array = 'explode ("carattere separatore", "stringa")';* dove, ipotizzando che la stringa da tokenizzare, contenuta nella variabile *$query, fosse stata: "*le piste carovaniere del deserto occidentale*" la sintassi della funzione sarebbe stata:
$array = 'explode (" ", "$query")'; ed avrebbe prodotto:
$array[1] ='lè,
$array[2] ='pistè,
$array[3] ='carovanierè,

Tramite la funzione predefinita *explode* sarebbe stato possibile, utilizzando un *array* come variabile d'appoggio, suddividere la stringa, parola per parola, in base ad un carattere separatore: il risultato sarebbe stato un *array* composto da *N* elementi dove *N* corrisponde al numero delle parole contenute nella stringa sottoposta alla funzione di *esplode*, ed avrebbe sopperito anche all'indicizzazione dei *tokens*; ciascun elemento dell'*array* avrebbe contenuto dunque una parola della *query*.

La *tokenizzazione* inizialmente presa in considerazione per implementare le opzioni di *ricerca libera-tutte le parole* e/o *ricerca libera-almeno una parola* della funzione di ricerca libera, divenne poi necessaria anche per la corretta esecuzione dell'estrazione dei campi *idi* dalla tabella *infor* per l'esecuzione di un processo di *I.R.* che corrispondesse ai criteri proposti dal *C.L.I.R.*, e sempre riguardo alla realizzazione di un sistema di *I.R.*, la *tokenizzazione* avrebbe costituito la base senza la quale non sarebbe stato possibile sottoporre la *query* ad alcun processo di *normalizzazione* e di *stemming*.[91]

4.5 eliminazione delle stopwords

In base a quanto affermato precedentemente sarebbe stato necessario implementare una procedura che intervenisse, dopo la tokenizzazione della *query*, e testasse per ogni *token*, se il suo valore corrispondesse a quello di una *stopword*.

La procedura avrebbe dovuto scorrere quindi tutto l'*array* contenente la stringa di *query* precedentemente *tokenizzata* ed eseguire il controllo relativo alla presenza o meno di *stopwords*; nel caso in cui esse fossero state trovate, allora avrebbe dovuto sopperire alla loro eliminazione dall'array.

La prima risorsa di cui la procedura di eliminazione delle *stopwords* avrebbe avuto bisogno sarebbe dunque una lista delle parole di stop in base alla quale eseguire tutte le operazioni successive.

La lista avrebbe dovuto essere composta per ciascuna lingua rappresentata nel sito, dato che la sua redazione monolingue avrebbe compromesso il funzionamento del *C.L.I.R.* precedentemente implementato.

La struttura del database bibliografiapiste sarebbe stata nuovamente ritoccata, con l'aggiunta di una nuova tabella, *stopwords* che avrebbe accolto le parole di stop.

La tabella avrebbe contenuto un solo campo, detto parole, all'interno del quale sarebbero state inserite tutte le parole di stop utili e necessarie al funzionamento della struttura.

$array[4] ='del',
$array[5] ='deserto',
$array[6] ='occidentalè

[91] L'implementazione delle procedure di *normalizzazione* e *stemming* sarà descritta successivamente, ma per entrambe i concetti è opportuno riferirsi alla *2° parte* dedicata all'*I.R.*.

Così come realizzato per la tabella infor, anche per la tabella *stopwords*, una volta raggiunto lo spazio massimo supportato per ogni record, sarebbe stato possibile aggiungerne uno nuovo, inserendovi degli aggiornamenti alle lingue già rappresentate, o magari inserendo la rappresentazione di un nuovo linguaggio:[92]

Parole
Italiano: *di,a,di,da,in,con,su,per,tra,fra,il,lo,la,gli,le,questo, codesto,, ecc…*
Inglese: *a,of,an,from,on,with,over,up,for,throught,across,,long, between,ecc…*
Portoghese: *a.bem,e,longe,para,se,você,abaixo,com,ela,mais, por,sem, ecc…*
Spagnolo: *a,aquí,cuantos,esta,misma,nosotras,querer,tales, usted,acá, ecc…*

La procedura, dunque avrebbe dovuto estrarre dalla tabella *stopwords* tutti i records, i cui valori sarebbero stati temporaneamente caricati all'interno di una variabile di appoggio.

La stringa di appoggio sarebbe anch'essa stata sottoposta alla funzione di *explode*,[93] e quindi *tokenizzata*, questo semplicemente per avere a disposizione un elenco indicizzato, parola di stop per parola di stop, al quale riferirsi con un semplice ciclo ricorsivo.

Infatti la struttura che avrebbe realizzato il confronto fra i tokens della *query* e quelli delle parole di stop sarebbe stato implementato con due cicli annidati[94] (dove cioè ogni giro del primo ciclo il secondo avrebbe compiuto tutte

---

[92] La lista di Stopword nelle varie lingue è tratta da *Oracle Text Reference Release 9.0.1*, reperibili presso il seguente *URL*: http://download-east.oracle.com/otndoc/oracle9i/901_doc/text.901/a90121/astopsup.htm

[93] In questo caso la funzione di *explode* non avrebbe avuto indicato, nella sua sintassi, il carattere separatore come blank (spazio vuoto, rappresentato da " "), ma il carattere che divideva una stopword dalla successiva sarebbe stato " , ", e dunque il codice corretto:
$arrayS = explode(",", $stopwords)
dove la variabile *$stopwords* avrebbe contenuto il valore dei campi parole della tabella *stopwords*,
l'*array $arrayS* avrebbe contenuto la *tokenizzazione* di *$stopwords* e, come precedentemente accennato, il carattere separatore per la creazione degli elementi dell'*array* sarebbe stato *","*.

[94] I due cicli annidati avrebbero permesso di testare, per ogni *token* della *query*, tutti i *tokens* di *stopwords*.
Date le variabili *i* e *j* rispettivamente indice del primo e del secondo ciclo,

le proprie iterazioni) i cui indici sarebbero stati usati per lo scorrimento degli elementi dei due array.

I *tokens* dell'*array* contenente la *tokenizzazione* della *stringa di query*, se fossero stati identificati con un *token di stop*, avrebbero assunto come valore la stringa " --- " ad indicare la cancellazione dell'elemento.

La procedura poi avrebbe ricostruito la *query*, sempre mantenendola fisicamente *tokenizzata*, escludendo quegli elementi il cui valore fosse stato cancellato.

4.6 stemming per normalizzazione della query

La procedura di normalizzazione avrebbe dovuto realizzare due processi principali:

- avrebbe dovuto attingere ad un archivio contenente suffissi e prefissi da ricercare all'interno dei tokens della *query*,

- se questi fossero stati presenti avrebbe dovuto procedere alla loro elisione dal valore del token del quale avrebbero costituito o la testa (se prefissi), o la coda (se suffissi o flessioni), o entrambe (una struttura della del tipo: *prefisso* - *tema o radice della parola* - *suffisso* o *flessione*)

---

le variabili *N* e *M* rispettivamente il *numero di parole della query* ed il *numero di tokens di stop*,
arrayQ ed *arrayS* rispettivamente gli *array* contenenti i *tokens* della *Query* e le *parole di Stop*
la sintassi sarebbe stata:

```
for ($i = 0; $i < $N; $i = ++$i) {
        for ($j = 0; $j < $M; $j = ++$j) {
                IF (strtolower($arrayQ[$i]) == $ arrayS [$j]) {
                $arrayQ[$i] = "---"; } // end IF
}       // end 2° for
} // end 1°for
```

La procedura eseguendo il codice citato avrebbe controllato, come da progetto, per ogni *tokens* di *query* se il suo valore fosse stato contenuto fra quelli estratti dalla tabella *stopwords*.
Se il test lo avesse verificato allora l'elemento dell'*array* corrispondente al *tokens di stop* avrebbe assunto il valore '---', ossia la procedura avrebbe letteralmente cancellato il valore dell'elemento arrayQ[$i].
La funzione *strlower* è una funzione di manipolazione delle stringhe predefinita di *Php*, e consente di riportare tutti i caratteri maiuscoli incontrati all'interno di una stringa ai corrispettivi minuscoli.
Anche in questo caso, così come per quanto espresso riguardo alle chiavi della tabella infor, sarebbe stato possibile applicare la procedura di stemming con lo stesso risultato di normalizzazione a query monolingue o multilingue.

Come la struttura implementata per l'eliminazione delle *stopwords* dalla tokenizzazione della *query* avrebbe dovuto riferirsi ad una tabella nel database bibliografiapiste contenente l'archivio delle diverse parole di stop, così questa procedura avrebbe dovuto riferirsi ad una ulteriore tabella contenente i prefissi ed i suffissi eventualmente da eliminare dalle parole della *query*.

La tabella sarebbe stata chiamata normalizzatore, e sarebbe stata organizzata in campi:

- *Suffissi*: contenente le flessioni grammaticali, le desinenze da apporre alla radice di una parola

- *Prefissi*: contenente le strutture da premettere al tema od alla radice della parola.

Entrambi i campi avrebbero dovuto comprendere prefissi e suffissi di tutte le lingue rappresentate all'interno del sito, e quindi la tabella normalizzatore avrebbe assunto la seguente architettura:

Suffissi:	*are,ere,ire,ato,ata,ano,ana,imo,imi,nte,ecc...*
Suffissi:	*ing, es,s,er,ed,an,/'s,ecc…*
Prefissi:	*stra, super, iper,ecc…*
Prefissi:	*over,never,ever,ecc…*

La struttura, come le tabelle realizzate precedentemente, avrebbe avuto un'elasticità tale da poter essere prima di tutto aggiornata per l'inserimento di nuove stringhe nei campi già realizzati, e poi per l'aggiunta di nuovi suffissi e prefissi di ulteriori lingue.

La struttura della tabella, inoltre, avrebbe consentito uno stemming con risultati corretti sia di stringhe monolingue (a prescindere da quale fosse stata la lingua di redazione) sia multilingue, mantenendo questa caratteristica già realizzata per la formulazione di *query* nel rispetto dei principi del *C.L.I.R.*

La prima parte della procedura di normalizzazione avrebbe dovuto provvedere all'estrazione dei prefissi e dei suffissi contenuti nei campi della tabella normalizzatore.

Questi sarebbero stati salvati temporaneamente in una stringa, poi sottoposta, come visionato per la procedura di *stopwords*, alla funzione di explode per ottenere una lista indicizzata delle strutture da utilizzare per la normalizzazione dei tokens di *query*.

La struttura di normalizzazione, allora, avrebbe dovuto procedere al processo di *stemming*: analizzare ogni *token* (precedentemente già passato dalla procedura di *stopwords*), ed identificarvi, se presenti, quei suffissi e quei prefissi aggiunti durante la corretta flessione grammaticale delle parole e presenti nell'array precedentemente generato dalla tokenizzazione dei contenuti della tabella normalizzatore tramite la funzione di *explode*.

Anche in questo caso sarebbero occorsi due cicli annidati per riuscire a controllare per ciascun token di *query* tutti i tokens di normalizzazione.

Una volta identificato il caso in cui la parola avesse contenuto in testa uno di questi prefissi, o in coda uno dei suffissi, il sistema avrebbe dovuto provvedere all'eliminazione delle parti superflue sostituendo all'interno dell'elemento in questione dell'array la parola priva della sua testa e/o coda.[95]

Sarebbe stato inoltre necessario, sempre nell'ordine di voler ricondurre ciascuna parola alla propria radice, controllare se la coda della parola già troncata, finisse per una vocale o per consonante.

Nel primo caso una ulteriore procedura avrebbe provveduto all'eliminazione della vocale, nel secondo caso la stessa avrebbe controllato se la consonante fosse stata una '*s*', associabile alla flessione inglese e spagnola del plurale, e quindi anch'essa da eliminare.

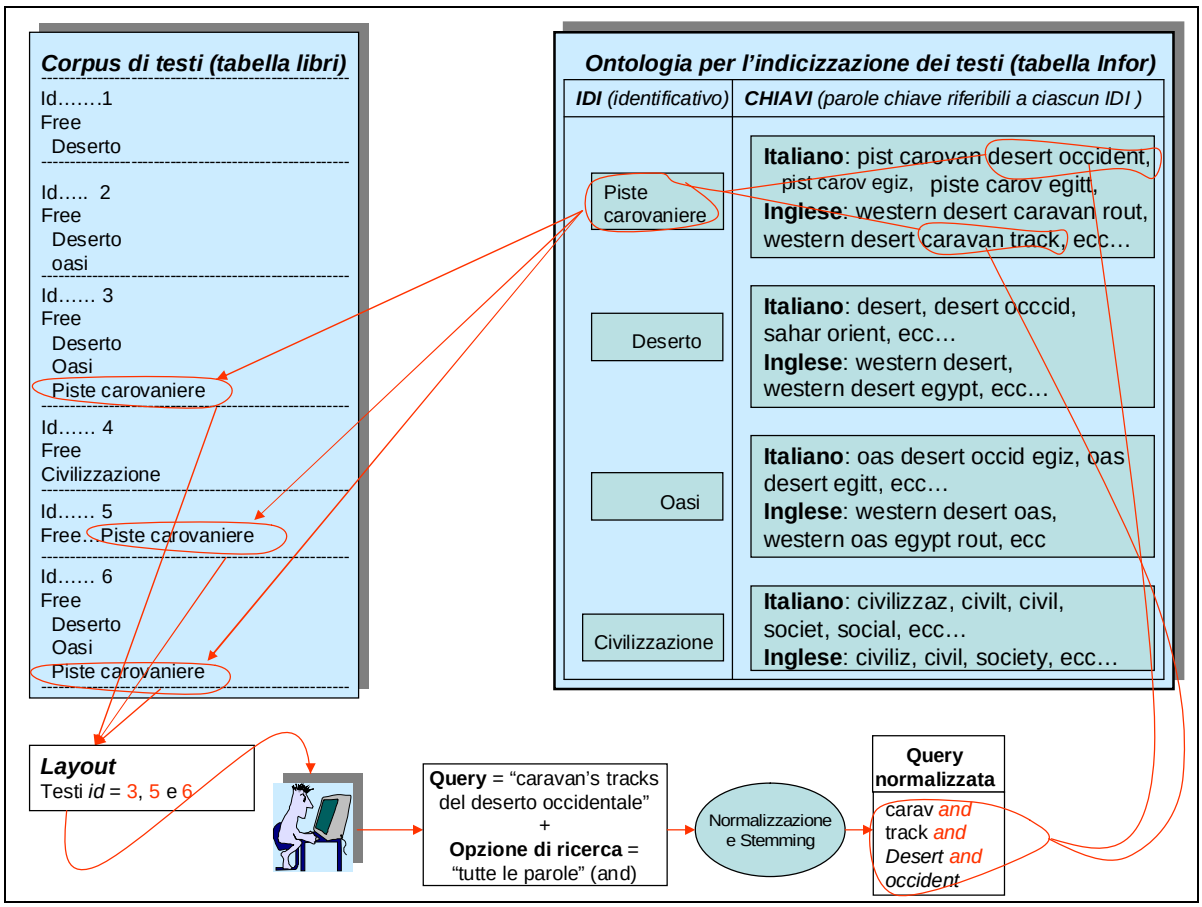
A questo punto la *query* iniziale era stata manipolata profondamente ed al massimo livello possibile (senza chiaramente intaccarne le informazioni trasportate e necessarie all'interrogazione verso il database).

IL sistema di recupero delle informazioni sarebbe inoltre stato capace di eseguire interrogazioni multilinguistiche al database proprio grazie all'architettura delle procedure di normalizzazione ed alla costruzione dell'ontologia Infor, capaci di essere estese in qualsisasi momento al supporto di un nuovo linguaggio.

---

[95] Infatti avrebbe potuto verificarsi il caso in cui la parola contenuta nel token avesse presentato sia un prefisso sia un suffisso normalizzabile.
Definendo con *P* la lunghezza del prefisso da eliminare, con *L* la lunghezza della parola e con *S* la lunghezza del suffisso da eliminare: Il troncamento della stringa in coda sarebbe avvenuta dal carattere di posizione *L – Z*, mentre l'elisione in testa avrebbe dovuto verificarsi dal *1°* carattere sino al carattere *P*.
Quindi della parola da normalizzare avrebbero dovuto essere considerati i caratteri *da N a (Z – P)*.

Sarebbe stato addirittura possibile formulare verso questo *C.L.I.R.S.* query contemporaneamente composte da parole in più lingue, come ad esempio: "*caravan's tracks del deserto occidentale*".



L'esempio di interrogazione è assolutamente improbabile, ma illustra a dovere le potenzialità del sistema di *C.L.I.R.*, che se implementato al supporto delle lingue in gioco, avrebbe realmente trattato e manipolato l'interrogazione ed il *Retrival* dei testi pertinenti in modo del tutto indipendente dalla query stessa di partenza.

5 III livello di MLIA: sperimentazioni multilinguistiche

Come da progetto, l'ultimo aspetto da considerare per completare la panoramica di questioni introdotte durante la discussione del *M.L.I.A.*, avrebbe consistito nell'affrontare la traduzione delle informazioni reperite dopo l'interrogazione.

La risorsa multilinguistica, per adesso, sarebbe stata limitata alla realizzazione di un sistema di *C.L.I.R.*, ma le informazioni estratte sarebbero state presentate nella lingua con cui erano state archiviate.

Riuscire a presentare, invece, i testi estratti, tradotti nella lingua madre dell'utente, o comunque da lui scelta perché a lui più vicina e comprensibile, sarebbe stato l'optimum del servizio bibliografico online.

Il momento della traduzione dei testi della bibliografia digitale sarebbe potuto avvenire in momenti molto diversi in dipendenza dal tipo di soluzione adottata per realizzarla, infatti il problema avrebbe potuto essere affrontato in due modi:

1. costruire una tabella *libri* per ciascuna lingua di cui si volesse rendere disponibile la traduzione dei testi,

2. trovare un sistema che, dopo aver ricevuto i dati estratti dall'archivio bibliografico tramite le strutture implementate, traducesse automaticamente le parti testuali contenute, ed inviasse il risultato al browser utente.

Il primo punto, il più semplice e probabilmente il più efficace e funzionante, avrebbe necessitato della traduzione di tutti i documenti contenuti nella bibliografia, e la loro redazione multilingue avrebbe dovuto essere stata disponibile prima ancora di creare la tabella libri.

Sarebbe stato necessario infatti affiancare alla tabella libri, contenente la scheda bibliografica di ciascun documento in italiano, ulteriori tabelle destinate e contenere la bibliografia tradotta, una per ciascuna lingua.

Un'altra soluzione, differente solo nel modo di organizzazione diverso, ma identica per spazio e risorse utilizzate, avrebbe consistito nel creare all'interno della tabella libri, tanti campi quante le lingue da rappresentare, contenenti la traduzione dei valori contenuti.

A seconda della lingua scelta dall'utente, in questo ultimo caso, sarebbero stati estratti i campi dei record archiviati nella tabella libri corrispondenti alla lingua selezionata, mentre nel primo esempio sarebbe stata indicata la tabella dedicata alla lingua scelta dall'utente.

Questi due tipi di soluzione, come affermato precedentemente, differenti nell'architettura e nell'organizzazione dei dati, ma identici per le risorse implicabili, avrebbero dovuto essere subito scartate, perché personalmente

impossibili da realizzare, data la mancata conoscenza delle lingue necessarie alla traduzione dei testi per poter costruire per ognuna di esse la traduzione corrispondente.[96]

La soluzione consistente nel redarre traduzioni parallele dei testi, vista la loro grandezza ristretta, avrebbe invece potuto essere adottata per quanto concerneva i menù e le informazioni scritte presenti invece nella parte grafica delle pagine web.

Infatti, oltre alla redazione multilingue dei contenuti della *bibliografia*, ed oltre alla risorsa offerta dal *C.L.I.R.*, sarebbe stato necessario rendere disponibile, almeno per le lingue principali, più parlate, tutto l'insieme di interfaccia grafica e testuale destinata al dialogo con l'utente, per ottimizzarne ancora di più la comprensione, e rendere chiare ed accessibili tutte le opzioni e tutte le caratteristiche disponibili dalle pagine *online*.

Questa operazione avrebbe invece potuto essere realizzata in tempi ristretti e con buoni risultati, utilizzando magari tutta quella serie di applicativi online o di software dedicati alla traduzione automatica ormai facilmente reperibili.

Inoltre il loro utilizzo avrebbe permesso la traduzione sia delle *stopwords*, sia delle *chiavi* contenute nell'*ontologia infor*, mentre la costruzione delle tabelle di suffissi e di prefissi destinati al processo di *stemming* per la *normalizzazione della query* avrebbero necessitato della consultazione e della conoscenza anche minima, ma necessaria, della grammatica della lingua presa in considerazione.

Il primo punto sarebbe dunque stato abbandonato per risolvere il problema a causa del quale era stato contemplato, ma sarebbe stato adottato per la risoluzione di altri problemi le cui caratteristiche gli si addicevano.

Abbandonato il primo punto, era necessario adottare la seconda soluzione, unica rimasta, seguendo la quale sarebbe stato necessario orientarsi nel vastissimo panorama riguardante la traduzione automatica che era già stato sviluppato precedentemente.

Prima di tutto sarebbe stato necessario stabilire in quale momento preciso del processo di *I.R.* avrebbe dovuto essere inserito il processo di *M.T.*.

Ovviamente le variabili da considerare per una giusta collocazione del processo sarebbero state quelle riguardanti l'efficienza del sistema, e quindi la velocità e la fedeltà della traduzione, che avrebbero dovuto essere entrambe le maggiori possibili.

Per quanto riguarda il rispetto della maggior fedeltà nella traduzione, il compito personale sarebbe consistito semplicemente nella cernita accurata di tutti i sistemi precedentemente descritti di *M.T.*.

---

[96] Questa sarebbe stata possibile se all'archivio bibliografico avesse lavorato uno staff costituito da personale e strumenti professionali capaci della traduzione dei testi redatti e nel controllo della fedeltà di quest'ultima, impossibile da realizzarsi nei tempi e nei costi progettati e nelle risorse reali disponibili.

Era necessario adottare cioè un sistema automatico di traduzione che riuscisse a redarre un testo nel linguaggio di destinazione che avrebbe dovuto contenere lo stesso contenuto semantico del testo sorgente, ed avrebbe dovuto esprimerlo in modo sintatticamente corretto per una sua adeguata comprensione.

Per quanto invece concerne il rispetto della seconda variabile dell'efficienza del sistema, il minor tempo possibile, sarebbe stato necessario ridurre al minimo le operazioni di *M.T.*, e dato che i processi a loro precedenti e successivi avrebbero riguardato solamente procedure di *I.R.*, l'unica quantità riducibile in grandezza sarebbe rimasta il numero di testi da sottoporre alla *M.T.*.

Esso avrebbe dovuto essere il minimo indispensabile, e quindi i testi tradotti sarebbero stati solamente quelli estratti dall'interrogazione: il processo di *M.T.*, a prescindere da quale o quali fossero stati adottati, avrebbe dovuto collocarsi dopo l'estrazione dei documenti dalla bibliografia digitale, e precedere la loro visualizzazione nel *browser*.

La soluzione avrebbe consentito al sistema di mantenere la propria efficienza di tempi di esecuzione di una sessione di *I.R.* e di spazio occupato, entrambi aspetti pesantemente intaccati dalla scelta precedentemente abbandonata, ossia la costruzione di tanti archivi bibliografici rispetto a quante lingue sarebbero state rappresentate.

5.1 Systran e BabelFish

L'idea di decodifica dei linguaggi naturali con tecniche automatiche si è trasformata in realtà ed obbiettivi di ricerca specializzata dopo la II Guerra Mondiale: durante gli anni 50, la ricerca sulla *M.T.*, sviluppò la maggior parte dei progetti e dei tentativi utilizzando calcolatori per le traduzioni *word-by-word*, senza usufruire di regole linguistiche.

Lo sviluppo non di tecniche adeguate di *M.T.* dunque non risultò ne rapido, ne facile ne poco dispendiose, e malgrado le difficoltà incontrate e le pause nella ricerca specifica la ricerca ha persistito soprattutto tramite enti economicamente finanziati dai governi, e quindi suscettibili alla mancanza di fondi non più ivi destinati per diversi motivi,[97] mentre Systran ha invece costituito uno del primi e pochi sviluppatori indipendenti restanti in ambito di M.T..

5.1.1 Systran

La tecnologia di *Systran* è stata sviluppata sotto il sistema operativo *Linux* ed è compatibile con tutte le diverse piattaforme: *Microsoft Windows*, *MACOS* ed *Unix*.

[97] Vedi *ALPAC*, ecc… nell'introduzione storica alla *M.T.*.

Tutti i diversi applicativi *Systran* utilizzano tecnologie di *Natural Language Processing*, *N.L.P.*, ed il sistema costituisce una risorsa di conoscenza multilingue utilizzabile tramite i calcolatori comuni.

I progressi raggiunti con *Systran* traggono il loro sviluppo proprio dagli applicativi di linguistica computazionale e di NLP incentrati a realizzare sistemi di *M.T.* compatibili alle applicazioni presenti sul web, per rendere disponibile un servizio di traduzione multilingue a strumenti quali la posta elettronica, Intranet, ecc…, e quindi mantenendo *Systran* attinente agli standard *Unicode* ed *HTTP*.

È necessario però affrontare nuovamente il problema della qualità della traduzione automatica anche per cercare di identificare la qualità del risultato ottenuto da *Systran*.

Prima di tutto è fondamentale distinguere fra sistemi capaci di una *M.T.* accurata e precisa, detta *M.T. personalizzata*, ed una traduzione meno puntuale, sintatticamente e semanticamente precisa se analizzata dettagliatamente, ma generalmente corretta, detta, appunto, *M.T. generica*.

È evidente che qualsiasi tentativo di *M.T.* dovrebbe tendere ad ottenere la migliore delle qualità possibili nella traduzione, ma questo obbiettivo, anche utilizzando i migliori dizionari e sistemi specializzati, non può sempre essere completamente soddisfatto ed è necessario ammettere che, almeno per adesso, non esistano ancora *M.T.S.* capaci di una traduzione impeccabile da un testo *sorgente* ad uno o più testi *target*.

Tuttavia è altrettanto necessario interrogarsi su quale delle seguenti due opzioni sia quella più adatta alle esigenze reali intrinseche ad un *M.T.S* fruibile sul web, e dunque scegliere tra l'effettiva necessità di una traduzione perfetta per adesso non raggiungibile, o adeguarsi ad un livello di traduzione meno esigente, ma generalmente corretta e comprensibile oggi raggiungibile con una certa gamma di applicativi tra cui anche *Systran*.

La tecnologia di *Systran*, oggi disponibile come risorsa dei maggiori motori di ricerca sul web, è una *M.T. multilingue generica* ed ha lo scopo di fornire una traduzione per la comprensione sommaria di un testo sorgente, avvalendosi comunque di una corretta *analisi sintattica* ed una *accettabile elaborazione semantica*.

Systran consente allora la produzione delle versioni nelle maggiori lingue presenti sul pianeta di un testo sorgente sintatticamente e semanticamente accettabili (nel dettaglio, in realtà non lo sarebbero…), e la sua azione corrisponde alla concreta facoltà dell'utente di poter osservare una pagina straniera comprendendone ugualmente le righe ed i contenuti principali trattati.

5.1.2 BabelFish

Come descritto precedentemente, *Babelfish* è uno tra i migliori *traduttori online* oggigiorno a disposizione degli utenti web.

Integrato da *Altavista* nel proprio portale, il servizio offerto da *Babelfish*, sviluppato con tecnologia *Systran*, comprende diversi tipi di traduzione: dal

gergo più colloquiale alla traduzione tecnica e nelle più importanti lingue del mondo.

Dall'*Inglese* è possibile tradurre direttamente in *Cinese*, *Giapponese*, *Coreano*, *Italiano*, *Francese*, *Spagnolo*, *Tedesco* e *Portoghese*, e viceversa, e quindi, ad esempio, anche se non direttamente traducibili testi dal *Cinese* all'*Italiano*, questi sono riconducibili all'*Inglese* per poi ottenere la versione italiana.

Altro servizio presente in *BabelFish* consiste nel fatto che *Altavista* offre la possibilità di inserire all'interno delle proprie pagine web uno script[98] che permette la visualizzazione di un *form* di traduzione, senza dover cioè rimandare l'utente alla pagina principale di *Altavista*/*BabelFish*,[99] e quindi senza il passaggio di testo (all'interno del *form* della pagina principale di *Altavista/BabelFish*[100] deve essere inserito il testo sorgente) e senza la necessità di selezionare il linguaggio sorgente (questo perché al momento della richiesta per poter usufruire dello *script BabelFish*, viene richiesto quale sia il linguaggio sorgente della pagina *web* nella quale comparirà lo *script*), ma semplicemente ciccando su una delle bandierine, una per ciascuna lingua supportata, per ottenere in tempo reale la traduzione della pagina stessa.

Form di BabelFish da inserire tramite script nella nelle proprie webpages

Il primo servizio europeo di traduzione automatica per *web Content* è stato lanciato nel 1997 dalla *Digital Equipment Corporation* e dalla *Systran S.A.* applicato e fisicamente ospitato dalla *Digital Altavista Search Site*.[101]

Per la prima volta, gli utenti avrebbero potuto tradurre le informazioni dallo standard linguistico inglese ad altre lingue in tempo reale e gratuitamente: il nuovo servizio ospitato sul *web Engine Altavista* avrebbe permesso la traduzione delle pagine web in 5 lingue europee: *Francese*, *Tedesco*, *Italiano*, *Portoghese* e *Spagnolo* da testi sorgenti in Inglese e viceversa; lo strumento di traduzione automatica venne chiamato *BabelFish*.[102]

Tramite *BabelFish* è possibile ottenere:

---

[98] Lo script, in Java, è reperibile compilando l'apposito form disponibile all'URL: http://www.altavista.com/sites/search/free_searchbox_transl

[99] http://babelfish.altavista.com/

[100] http://world.altavista.com/sites/itit/pos/babelfish/trns

[101] http://babelfish.altavista.com/

[102] Il nome è stato ispirato al best seller *The Hitch Hiker's Guide to the Galaxy*, scritto da Steve Meretzky e Douglas Adams, nel quale gli autori ipotizzano un dispositivo di traduzione automatica denominato, appunto BabelFish, Pesce di Babele, dispositivo che avrebbe dovuto essere direttamente inserito nell'orecchio ed avrebbe interfacciato direttamente la traduzione automatica.

- *Translating Raw Text* (traduzione diretta dei testi): collegandosi al sito di traduzione di *Altavista*,[103] l'utente ha la possibilità di inserire il testo sorgente nell'apposito spazio del *form*; l'utente poi deve selezionare sia la lingua sorgente, sia la lingua target della traduzione del testo inserito.

- *Translating webpages*: il *form* per gestire le opzioni di traduzione offerte da *BabelFish* permette di inserire l'*URL* della pagina web che ospita il testo da tradurre. Anche in questo caso è necessario selezionare linguaggio *sorgente* e linguaggio *target*, e d anche in questo caso si ottiene sia la visualizzazione del testo tradotto sia quella del testo *sorgente*.

- *Translating Search Result*: il *web Engine Altavista* offre, dopo aver effettuato un'interrogazione di ricerca ed aver ottenuto di risultati, di tradurne la sommarizzazione per poterla rende comprensibile fuori dallo standard della sua redazione.

Il livello di traduzione ottenibile con *BabelFish* è accettabile se considerate le caratteristiche e le aspettative di comprensione e di *M.T.* generale illustrate per *Systran*, e malgrado i difetti evidenti nella produzione di testi in linguaggi target semanticamente sintatticamente corretti, questa risorsa è indubbiamente interessante e coinvolgente particolarmente per quanti siano direttamente implicati ed addetti alla comunicazione tecnica internazionale.[104]

5.2 sperimentazioni con Babelfish

Il livello di traduzione, così come esaminato precedentemente, è certamente di bassa qualità, soprattutto per determinate lingue, dove spiccano evidenti errori sintattici e semantici che si concretizzano in periodi improbabili e grandi difficoltà di disambiguazione dei termini.

Tuttavia questa non è una valutazione del sistema inaspettata, infatti era chiaro ed evidente che il sistema di *M.T. BabelFish*, avrebbe prodotto una traduzione generale,[105] ma che con i suoi difetti, sarebbe stata sufficientemente

[103] http://www.altavista.com

[104] *Babelfish* comprende diversi tipi di traduzione: dal gergo alla traduzione tecnica, il tutto nelle più importanti lingue del mondo. Come descritto precedentemente dall'inglese è possibile tradurre in Italiano, Francese, Spagnolo, Tedesco e Portoghese, ma è altresì possibile tradurre termini e testi scritti in Russo, Giapponese e Cinese direttamente in Inglese, per poi avviarne la traduzione nelle altre lingue.

[105] Vedi 3.5.1. a proposito dei livelli di traduzione e delle caratteristiche di *Systran*

chiara da permettere all'utente una comprensione accettabile dei contenuti espressi.

Il link al sistema di traduzione di *Babelfish* avrebbe dovuto avvenire in modo trasparente, e produrre a seconda della versione del portale scelta, la traduzione dei contenuti delle tre pagine precedentemente elencate, corrispondente alla lingua scelta inizialmente dall'utente.

Sarebbe stato dunque necessario inviare a *BabelFish*, come *URL* relativo al testo da tradurre, l'*URL* corrispondente di ciascuna voce di un indice puntato, dalla quale poter accedere alla visualizzazione della scheda relativa o ad un documento della bibliografia.

Nel momento in cui l'utente avesse selezionato una delle voci dell'elenco puntato, il *link* non sarebbe dunque più stato diretto con la pagina *doc.php*, ma avrebbe inviato a *BabelFish* l'*URL* corrispondente con i giusti parametri di traduzione, ed avrebbe visualizzato la pagina tradotta nella lingua target prescelta in modo invisibile all'utente.

Il funzionamento stesso di *Systran* e *BabelFish* avrebbe imposto che il documento *sorgente*, per ottenere una traduzione diretta nelle altre lingue, così come descritto precedentemente a proposito dell'architettura e del sistema *Systran*, fosse stato in *Inglese*, l'unica lingua dalla quale sarebbe stato possibile risalire a *Spagnolo*, *Portoghese*, *Francese*, ecc…

Momentaneamente applicato solo ai documenti archiviati all'interno della bibliografia, il sistema di *M.T.* tramite *Systran* e *BabelFish* avrebbe reso necessario creare una seconda tabella, chiamata *Book*, identica alla tabella libri, e contenente una versione inglese delle schede archivianti tutti i documenti della *bibliografia*.[106]

Ogni indice puntato ad essa riferito, avrebbe dovuto essere modificato nel codice generante il link[107] precedentemente diretto a *doc.php*, adesso riferito a *BabelFish* per la traduzione di *doc.php* nella lingua prescelta.[108]

---

[106] Una descrizione molto più approfondita circa l'architettura della tabella *Book* sarà fornita durante la descrizione della sperimentazione di *UNL* applicato al portale.
La tabella *Book*, infatti, era stata inizialmente progettata solo per la sua applicazione ad *UNL*, ma poi presentandosi la possibilità di fornire un servizio multilingua accettabile tramite il sistema di *M.T.* offerto da *BabelFish* e *Systran*, la tabella *Book* è stata utilizzata proprio come archivio dei testi sorgente per le varie versioni multilinguistiche dei documenti della bibliografia.

[107] Mentre prima ad ogni indice puntato corrispondeva un link dal seguente codice:
<a href = doc.php?id=$id> dove la dinamicità del link (l'*URL* a cui si riferisce cambia per ogni elemento della tabella) è data dalla variabile $id, e generante un link diretto alla pagina doc.php relativa all'elemento con campo id del valore di $id (parte del codice finale: *doc.php?id=$id*), in questo caso era necessario creare un nuovo tipo di *link* dinamico, che mantenesse la sua relazione col campo id ed il valore della variabile $id per riferirsi al record selezionato dall'indice, ma che nella parte precedente del codice avesse le giuste coordinate per linkarsi a *BabelFish* ed inviare l'*URL* della pagina desiderata più i parametri della traduzione (in quale lingua *BabelFish* avrebbe dovuto tradurre il documento sorgente in Inglese contenuto dall'*URL*), il codice sarebbe stato:

Il sistema così creato, non dipendendo per quanto concerne il panorama delle lingue supportate da risorse implementate o meno all'interno del portale, sarebbe stato automaticamente aggiornabile a nuovi linguaggi nel momento in cui il servizio stesso di traduzione fosse stato implementato all'interno di *BabelFish*.

È quindi stato così possibile dotare il portale di un'ulteriore risorsa per una consultazione approfondita delle risorse bibliografiche.

5.3 U. N. L., Universal Networking Language

In seno alla *United Nations University*[109] nel 1995 è stato fondato *l'Institute for Advanced Studies* (*I.A.S.*),[110] un istituto avanzato di ricerca ed educazione con un

---

<ahref=\http://babelfish.altavista.com/tr?doit=done&urltext=http://membres.lycos.fr/bibliografiapiste/biblioengl/doc.php?id=$id&lp=en_es\>
Malgrado il codice sia estremamente prolisso, è abbastanza ben leggibile, e ci permette di identificare una prima parte, in azzurro, contenente le coordinate del *server* di *M.T.* di *BabelFish*, e nella seconda, in rosso, l'*URL* relativo al documento con campo *id = $id* e le impostazioni di traduzione, da *Inglese* a *Spagnolo* (*lp=en_es*).

[108] La lingua è prescelta perché l'utente al momento del collegamento con la prima pagina del portale deve scegliere la lingua con cui entrare, interfaccia che poi sarà presente per tutto il sito, e che sarà impostata come linguaggio target delle traduzioni automatiche incontrate durante la consultazione delle risorse.

[109] Nel 1969 durante l'introduzione al *Rapporto Annuale* dell'*Assemblea Generale delle Nazioni Unite*, il *Segretario Generale* suggerì che si fosse ormai giunti ad un momento adatto alla seria considerazione di stabilire una *Università Internazionale*. Egli pervenne a questa conclusione perché la sua attenzione venne attratta da singoli lavori eseguiti nell'intento di stabilire delle istituzioni di istruzione a carattere internazionale.
Questa *Università Internazionale* Avrebbe perseguito obbiettivi quali il progresso mondiale e la pace, sarebbe stata organizzata con docenti di molte nazioni e di tutto il mondo, e sarebbe servita come esempio e promotrice dell'abbattimento delle barriere quali l'incomprensione e la diffidenza fra nazioni e culture diverse.
Lo scopo primario dell'*Università Internazionale*, così come espresso dall'allora Segretario Generale, sarebbe stato quello di promotrice del dialogo e della comprensione internazionale tanto a livello politico quanto a livello culturale, e la sua organizzazione pratica sarebbe stata compito *dell'UNESCO, United Nations Educational, Scientific and Cultural Organization*, che avrebbe anche ricevuto la responsabilità per la progettazione e la realizzazione nel dettaglio dell'università stessa, selezionando lo staff amministrativo e nominando alla guida di essa uno studioso di fama mondiale; l'università avrebbe dovuto essere localizzata in un paese noto per il suo spirito di tolleranza e per la libertà di pensiero: il *Segretario Generale* espresse la propria speranza che *l'UNESCO* trovasse possibile lo sviluppo dell'idea suggerita.
Il 13 Dicembre del 1969 il *Consiglio Generale* approvò l'iniziativa ed il *Segretario Generale* venne invitato a produrre, in cooperazione con *l'UNESCO*, con lo *United Nations Institute for Training and Research*, ed interpellando qualsiasi altro ente fosse ritenuto necessario, e naturalmente tenendo in considerazione l'opinione espressa dal *Consiglio Generale*, uno studio competente circa l'effettiva realizzazione *dell'Università Internazionale*, includendo chiare definizioni delle mete proposte e degli obbiettivi da intraprendere, includendovi anche i metodi di finanziamento previsti.

orientamento flessibile e multitematico verso le interazioni fra i sistemi sociali e naturali.

In questi suoi primi anni di vita l'istituto ha rivolto i suoi sforzi sullo studio dello sviluppo sostenibile.[111]

È proprio nell'ambito dell'*U.N.U./I.A.S.*, durante la direzione del dr. *Tarcisio Della Senta*, oggi direttore della *Fondazione U.N.D.L.* con sede a *Ginevra*, che è stato concepito da due ricercatori giapponesi nella prima metà degli anni novanta il progetto *UNL*.

Il dottor *Kazuhiko Nishi, visiting professor* dell'*I.A.S.*, esperto in sistemi di mezzi di comunicazione, e il dottor *Hiroshi Uchida*, da anni nel vivo della traduzione automatica, si trovarono infatti concordi sulla realizzabilità di un

---

Il *Consiglio* espresse la speranza che il rapporto sarebbe stato reso presto disponibile durante *l'Anno dell'Educazione Internazionale* (*International Education Year*), così da poter essere sottoposto al *Consiglio Economico e Sociale* della *Riunione Generale* del 1970. Le decisioni dell'assemblea furono ratificate nella *Risoluzione* n. *2573* (*XXIV*), ed il rapporto al progetto fu proposto al *Consiglio Economico e Finanziario* che lo approvò senza alcuna obbiezione il 4 Dicembre del 1969.

L'UNU (*United Nations University*) fu fondata durante il *Consiglio Generale* del 1973, istituita come una comunità di studiosi dediti alla ricerca, all'istruzione avanzata, alla diffusione della conoscenza inerente problemi globali e ponderanti per la sopravvivenza, lo sviluppo, la salvaguardia dell'uomo. *L'UNU* avviò la propria attività nel 1975 presso la sua sede centrale a *Tokio*: le sue attività si concentrano soprattutto sulla pace e le soluzioni dei conflitti, sullo sviluppo mondiale della scienza e della tecnologia in relazione alla salvaguardia dell'uomo.

L'università opera attraverso una rete mondiale di centri di ricerca, d'istruzione e di perfezionamento pianificati e coordinati dalla sede centrale di *Tokio*.

[110] La sede dello *UNU/IAS* è adiacente al centro della *UNU*, nel cuore di Tokio in *Omotesando*, nell'edificio che fu offerto generosamente dalla *Tokyo Metropolitan Government* e fu inaugurato nel Novembre del 1995, adesso uno dei più moderni centri della rete *UNU*.

Avvalendosi di esperti internazionali, così come dell'apporto culturale di ciascun paese, *UNU/IAS* è impegnato nello studio avanzato di soluzioni creative inerenti ai problemi incalzanti di interesse globale. Il termine "*studio avanzato*" si riferisce ad un approccio alle diverse questioni con metodi multidisciplinari: *l'UNU/IAS* impegna esperti di discipline tradizionali come economie, legge, biologia, scienze politiche, fisica, chimica, informatica, ecc…, chiedendo loro di condividere la propria specifica conoscenza in un tentativo focalizzato alla comprensione ed al chiarimento di talune delle chiavi per la promozione allo sviluppo.

Avvalendosi inoltre di metodologie di ricerca avanzate ed approcci con soluzioni creative ai problemi globali e sempre più incalzanti, l'attività di ricerca *dell'UNU/IAS* è progredita puntando continuamente all'obbiettivo concreto di identificare le aree strategiche di ricerca per la risoluzione dei maggiori problemi per l'umanità, interagendo con governi ed organi decisionali, particolarmente, per i paesi in via di sviluppo.

[111] Le aree in cui è attualmente attivo sono: *Eco-ricostruzione*, *metropoli* e sviluppo urbano, multilateralismo e governo, scienza, tecnologia e società.

sistema basato sullo sviluppo di un linguaggio elettronico astratto[112] per la rappresentazione di testi redatti in linguaggio naturale.

Il programma *UNL* riprendeva e sviluppava l'antica idea di poter esprimere il significato di un'espressione in linguaggio naturale (dunque anche di un testo) per mezzo di una *grammatica universale*, e di poter "tradurre" questa rappresentazione in qualsiasi linguaggio naturale, concetto che è alla base anche *dei sistemi ad interlingua*.

UNL è stato promosso e sostenuto all'interno di organismi delle *Nazioni Unite* con lo scopo di abbattere la barriera dello "*standard linguistico*" su web, possibile generatore di discriminazioni culturali, poiché fuori di esso, ossia la popolazione incapace di parlare e comprendere *lingua inglese*, risiede circa il 75% della popolazione mondiale.

Essenzialmente, questo *linguaggio elettronico* verte sull'analisi dei concetti presenti nelle frasi che costituiscono un testo e delle relazioni logiche che si stabiliscono tra di essi.

Il compito di rappresentare i concetti è affidato a "*parole universali*" espresse con parole del vocabolario inglese "ristrette" per mezzo di vincoli che ne limitano la polisemia ad un significato preciso e che sono organizzate in una Base di Conoscenza tassonomica.

---

[112] UNL è stato concepito come un linguaggio elettronico per computer mirato per esprimere conoscenza, dunque non esattamente un'interlingua. In una delle sue molteplici applicazioni può essere usato come un'interlingua. (bisogna ricordare che M.T. è solo un'applicazione possibile di UNL). Lo stesso Uchida sostiene: "It should be reminded that the fundamental premises that the UNL is an electronic language for computers to express knowledge.
UNL is supposed to be used as a way for representing the world, for storing and transmitting information, for communicating, for exchanging experiences and knowledge, for expressing feelings and thoughts, for organizing the way of thinking, for interacting. The only difference between UNL and NLs concern the user: UNL is to be used by computers instead of humans. As a consequence, UNL is committed not to allow for ambiguity and vagueness.
UNL is also an autonomous (independent) language. UNL borrows signs from English, but it is not dependent on English or on any other NL. UNL is not what languages have in common, it is not the intersection between all existing languages, it is not the underlying structure present in every NL. UNL is not a metalanguage.
UNL is rather a language containing ways of referring to all the knowledge referred to by all existing languages. It does not comprise only ways for referring to universal references, depicted and referred to by every language. It comprises also ways for referring to every particular reference portrayed by every particular language. UNL consists of signs (Universal Words, Relations, Attributes) coping not only with the shared world or the shared knowledge of world, but also with the very idiosyncratic way through which each language categorizes and organizes its world experience.
It encloses all conceptual categories needed to be coindexed to each NL. Regarding coverage and descriptive power, UNL is supposed to be a natural-like artificial language.
In this sense, UNL is not an Interlingua. It should be used neither as an intermediate (semantic) representation between NLs nor as a controlled (disambiguated) subset of English. If UNL is to be used in machine translation systems, it should play the role of a source or a target language instead of some intermediate position. That means that UNL may be used as an Interlingua. "

La posizione della *parola universale* nella tassonomia stabilisce la sfera concettuale alla quale appartiene, in altre parole, il concetto che denota.

UNL fa uso anche di un insieme di attributi che aggiungono informazioni sull'uso del concetto nell'ambito della frase, e questi attributi rivelano in che modo chi parla o scrive si pone nei confronti dei concetti che esprime.

Le relazioni tra i concetti espressi dalle "*parole universali*" insieme agli attributi costituiscono gli enunciati di *UNL*.

Tale tipo di lingua, a differenza di una lingua artificiale come l'esperanto, non è leggibile da un umano ma solo da un software, ed in questo senso la si può definire *una lingua elettronica*.

Ammettendo che la grammatica di una qualsiasi lingua naturale è troppo complicata ed estesa per essere interpretata nell'analisi da parte di una macchina in maniera veloce e non ambigua, di fronte a questa difficoltà appare adeguata la soluzione di adottare una "*lingua franca*" che faccia da intermediario fra una lingua sorgente e un numero a volontà di lingue di arrivo.

Una volta codificato un qualsiasi testo in questo *meta-linguaggio*, è possibile ricavarne la *deconversione* in una lingua naturale a scelta.

Con questo linguaggio, chiamato *Universal Networking Language*, si mette in evidenza il significato profondo del testo e non la sua strutturazione sintattica, un concetto diverso rispetto alla traduzione automatica classica, la quale si pone a valle delle lingue naturali, eseguendo programmi di traduzione *uno-a-uno* pur passando attraverso *un'interlingua* ad hoc per la coppia di linguaggi da gestire.

Il sistema *UNL*, invece, anticipa il problema, sostituendo i documenti espressi in varie lingue naturali con testi scritti in questo *meta-linguaggio* molto generale da cui, in seguito, produrre documenti reali.

Il progetto ha cercato di sviluppare e promuovere una piattaforma di comunicazione multilingue, con lo scopo di abilitare ogni popolo alla condivisione delle informazioni e della conoscenza nella loro lingua madre.[113]

Il sistema non raggiunge certamente vertici di raffinatezza linguistica, ma può essere ben applicato al settore delle informazioni tecnico-scientifiche, dei servizi e del commercio via Internet, e a tutte le applicazioni dal linguaggio settoriale con ottimi risultati.

5.4 componenti e specifiche del sistema UNL

[113] Ovviamente lo sforzo per il successo di un tale sistema è enorme, specie se consideriamo che (per la prima fase del progetto) si è previsto lo sviluppo di moduli per 18 lingue naturali: le 6 lingue ufficiali *dell'ONU* (Arabo, Cinese, Inglese, Francese, Russo, Spagnolo) più Greco, Tedesco, Tailandese, Malese, Coreano, Giapponese, Portoghese, Lettone, Indonesiano, Mongolo, Hindi ed Italiano (il cui programma per la traduzione è opera *dell'Istituto di Linguistica computazionale del Consiglio Nazionale delle Ricerche di Pisa*, coordinato dal Professor *Antonio Zampolli* sin dalla partenza del progetto stesso, nel 1996 e diretto dalla Professoressa *Irina Prodanof*.

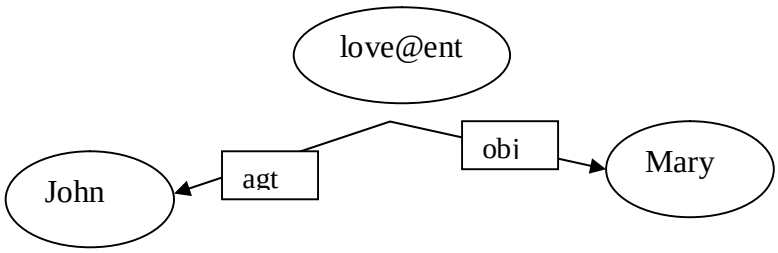
Il sistema *UNL* consiste in un linguaggio destinato al computer, e costruito per rappresentare ed esprimere qualsiasi tipo di contenuto .

Esso è stato concepito per rappresentare un testo frase per frase come un grafo i cui nodi sono i *concetti*, mentre gli archi rappresentano le *relazioni* tra gli stessi *concetti*: ognuno dei *concetti* viene espresso tramite una *parola universale* (da qui *UWs, Universal Words*).

Il contenuto di una frase è dato da *Uws* con le relazioni binarie che si stabiliscono tra di loro: la radice del grafo è rappresentata dalla UW con attributo @entry.

15. John loves Mary

agt(love.@entry.@present, John)
obj(love.@entry.@present, Mary)



Quindi ogni frase è costituita da un grafo , ed essa consiste nell'insieme delle *relazioni binarie* e *dirette* fra due *concetti* (*nodi* del grafo): la relazione binaria è visibile come l'unità base, il mattone della struttura *UNL*.

(non ho capito cosa volevi dire con la frase di prima) Un documento *UNL* può essere definito *una serie di relazioni fra concetti in formato HTML*.

Per descrivere la relazione logica espressa da ciascun arco fra due UWs sono state specificate diversi tipi di relazioni che possono verificarsi tra tipi di concetti della Base di Conoscenza (e.g. la relazione instrument si stabilisce tra un verbo di azione ed un oggetto concreto ins(do, concrete thing) mentre final state si può stabilire tra un verbo nella tassonomia di *processo* ed un concetto nella tassonomia di *oggetto*) gol(occur(gol>thing), thing) oppure gol(do(gol>thing), thing).

La stabilità e la coerenza del linguaggio *UNL* viene assicurata proprio dalla Base di Conoscenza rispetto alla quale si specificano le relazioni tra concetti espressi dalle Parole Universali contenute nel Dizionario di Parole Universali indicate all'interno del sistema da diverse etichette: ad esempio la relazione che specificherà il rapporto fra l'agente e l'azione compiuta sarà diversa dalla relazione che descrive il rapporto fra un evento ed il luogo, oppure il tempo in cui l'azione si svolge:

La struttura interna di ciascuna relazione è invece costruita secondo questo criterio: [114]

13. <Relazione> ::= <Sigla relazione>
[":" <Composta da UW-ID>] "(" { <UW1> ":" <UW-ID1> } |
":" < Composta da UW-ID1> } "," {<UW2> ":" <UW-ID2> } |
":" < Composta da UW-ID2> } ")"

Questo insieme di relazioni, tuttavia, descrive solamente degli eventi, i processi e gli stati obbiettivi, e non riescono ad esprimere tutta quella sorta di informazioni aggiuntive come gli atti linguistici (chi parla sta informando, comandando, richiedendo qualcosa, invitando qualcuno, ecc…), le attitudini (avere intenzione di…, augurarsi che…, aspettarsi che…, …essere costretti a…, ecc…), e tutte quelle connotazioni che riusciamo a individuare in una frase, come se essa, attraverso i significanti (in una lingua comune serie di parole scritte o pronunciate rispondenti ed organizzate secondo determinate leggi grammaticali, convenzioni, ecc…, mentre in *UNL* rappresentate da relazioni fra UWs) recasse con se tutta una serie di qualità e di sfumature contemporaneamente necessarie alla comprensione ed ulteriori al concetto della frase stessa.

Ogni lingua adotta dei codici propri, delle soluzioni diverse per esprimere tutta questa gamma di informazioni aggiuntive appena citate, che invece *UNL* deve riuscire ad esprimerle in modo univoco, data la sua premessa di universalità.

Proprio perché la sua caratteristica di Universalità non è un risultato a posteriori, ma una premessa del sistema, dunque il fatto che diverse lingue adottino diversi codici per le particolari espressioni che un parlante può aggiungere alla sua comunicazione orale o scritta è del tutto indifferente, e queste informazioni saranno rappresentate tramite degli attributi, che nella

---

[114] Legenda:

- i simboli < (apertura di un *Tag*) *e* > (chiusura di un Tag) indicano una parola non tecnica, variabile,
- le coppie di parentesi quadre, *[ ]* includono al loro interno elementi opzionali,
- le coppie di parentesi graffe, *{ }* includono elementi alternativi,
- gli apici, *“ ”* , includono al loro interno una stringa,
- così come in quasi tutte le sintassi di programmazione, il simbolo | rappresenta l’operatore logico “*or*”, e cioè una disgiunzione,
- il simbolo iniziale ::= indica che l’elemento alla sua sinistra è definito come l’espressione alla sua destra (così come in C++, Php, Pascal, ecc… la successione dei caratteri *:=*, o *:= =* indica un assegnamento di valore, che può essere interpretato, data l’espressione *a := f(b)* come *a* assume il valore di, e cioè è definito come *f(b))*

sintassi del sistema saranno marcati dal carattere @ (at), accompagnando i concetti in relazione fra loro.

Per esempio il concetto di tempo rispetto ad un soggetto agente potrebbe essere anteriore, contemporaneo e posteriore, e questa informazione sarà indicata nella relazione <*act*> fra soggetto agente ed azione compita dall'attributo *@past* se ormai è un'azione compiuta, avvenuta nel passato, *@present* se l'azione è contemporanea al tempo del soggetto parlante, *@future* se l'azione sarà successiva. La serie di attributi nel sistema possono essere suddivisi in 7 gruppi principali:

1. *attributi generali,*
2. *attributi che esprimono l'attitudine ed emozioni del narratore,*
3. *Attributi che indicano i punti di vista del parlante,*
4. *attributi che enfatizzano una o più parti del discorso,*
5. *attributi che descrivono il tempo verbale,*
6. *attributi che indicano l'aspetto,*
7. *attributi che precisano l'oggetto a cui si riferisce l'evento*

Mentre la lista delle relazioni è una lista chiusa, quella degli attributi è aperta e può incrementare man mano che altre lingue con altre caratteristiche linguistiche particolari si aggiungono alla comunità *UNL*.

5.5 funzionamento di UNL

Enconversione, *deconversione*, *editor* e *viewer UNL*

- *Enconversione* (codifica in *UNL*), *deconversione* (decodifica dell'espressione *UNL* in un linguaggio naturale): sono essenzialmente due strumenti, un parser ed un generatore, capaci di codificare un linguaggio naturale in espressioni *UNL*, ed a sua volta capaci di ricondurre un'espressione ad un linguaggio naturale l'espressione precedentemente elaborata: questi due processi sono detti rispettivamente *Encodifica* e *Decodifica*.
  Entrambi gli strumenti insieme alle risorse linguistiche specifiche per una lingua (dizionari di lingua e grammatiche per l'analisi e la generazione) costituiscono

un *Language Server*, e riceve le informazioni da rielaborare attraverso il web.

- *Editor* e *Viewer UNL* per la manipolazione e la visualizzazione delle espressioni in *UNL*

Il *Language Server* si compone di:

Risorse:

- *Dizionario di lingua* (parole della lingua e corrispettivi concetti in Uws che esprimono)
- Una grammatica per *Enconversione*
- Una grammatica di *deconversione*

Software:

- *Enconverter*
- *Deconverter*

Encodificatore: *l'encodificatore*

È un software che, automaticamente o interattivamente, riesce a trasformare un testo espresso in un linguaggio naturale alla sua rappresentazione in *UNL*, e cioè trasforma il testo da linguaggio naturale ad espressioni *UNL*: il software capace di eseguire *l'encodifica* è stato sviluppato dall'istituto *UNU/IAS*, ed è stato chiamato *EnCo*.

La caratteristica prima di *EnCo* è la sua universalità, e cioè la su applicabilità ad ogni linguaggio naturale, ed è affiancato nel suo lavoro di *encodifica* da un *dizionario* di parole, un *dizionario* di occorrenze e, naturalmente, dalla *regole di conversione* per la determinata lingua.

È proprio tramite i *dizionari* e le *regole di conversione* che *EnCo* riceve le informazioni necessarie a manipolare il documento in linguaggio naturale ed a trasformarlo in *UNL*, perché i *dizionari* a cui si appoggia contengono informazioni come il tipo di *UW*, le parole del linguaggio usate per esprimere suddetta *UW*, le loro proprietà sintattiche, ecc...

EnCo, inoltre, riesce a produrre documenti in *UNL* senza che l'utente abbia la minima conoscenza della sua sintassi e delle sue regole.

Deconvertitore

È un software che riesce a ricondurre espressioni *UNL* in frasi espresse con un linguaggio naturale, e costituisce la parte più consolidata del processo di traduzione automatica.

Piu' delicata è la fase di Enconversione, data l'ambiguità' insita nel linguaggio naturale: l'espressione *UNL* non deve essere ambigua, perché questo determinerebbe la generazione di piu' soluzioni in Deconversione, fatto tecnicamente di difficile gestione e che potrebbe portare a errori.. L'espressione in *UNL* deve essere, perciò, unica, non ambigua.

Per questo, spesso, la fase automatica di Enconversione viene sia preceduta da una fase di pre-processing, sia seguita da un post-processing manuale per disambiguare laddove si ottengono da una frase in LN piu' soluzioni in *UNL*.

Un'altra strategia adottabile potrebbe consistere nel precedere la fase automatica da un pre-processing che disambiguasse le parti della frase, come nell'esempio precedente "man in the park with the telescope": in *UNL* la frase darebbe 2 soluzioni rispettivamente con relazioni diverse; una di esse deve essere eliminata.

Il software sviluppato dall'istituto *UNU/IAS* è detto *DeCo*, anch'esso come *EnCo* equipaggiato con dizionari di lingua , di occorrenze e delle regole di deconversione per il linguaggio rispettivo, ed è anch'esso caratterizzato dall'applicabilità a tutti i linguaggi naturali.

Editor e Viewer

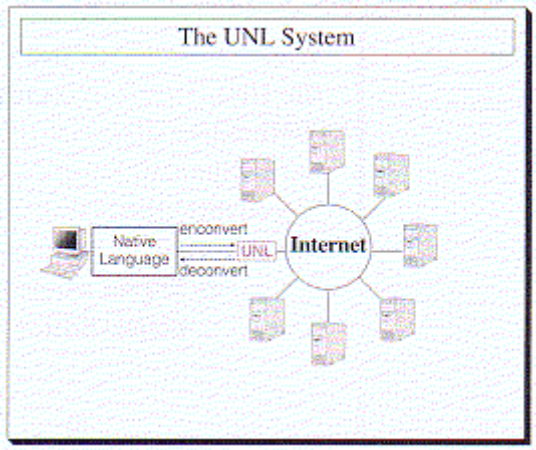
Per creare dei documenti in *UNL* è necessario un editor che sia collegato ad un *Language Server*, munito di *EnCo* e *DeCo* per determinati linguaggi naturali: *l'editor* invierà al *server* i documenti scritti nel linguaggio naturale, il *server* rielaborerà il testo in LN in un documento *UNL* che tramite il Viewer sarà mandato ad un altro Language Server per essere deconvertito…. ed invierà al *viewer* il documento decodificato nei diversi linguaggi.

Durante questo processo le espressioni in *UNL* possono essere prodotte automaticamente oppure l'autore potrà interagire: difatti vi sono 4 tipi di editor *UNL* che, a seconda del metodo di *enconversione*, sono *completamente automatizzati* (full automatic enconversion for natural language texts) o ad un livello di interattività sempre maggiore (full automatic enconversion for controlled or tagged language texts, interactive enconversion for natural language texts, word by word input method).

L'interfaccia *utente-server* nella fase ultimata delle decodifica è il *viewer* ed esso riceve il documento ormai decodificato dal *Language Server*: il documento è quindi tradotto nel linguaggio finale.

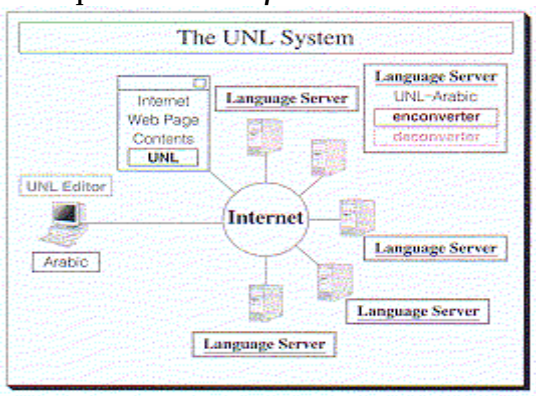
Il funzionamento del sistema *UNL* si basa dunque su una rete di *Language Server* interconnessi fra loro ed ai quali possono accedere gli *editor – viewer* di documenti *UNL*.

L'editor UNL riconosce la lingua, ad esempio l'Arabo, in cui è redatto il documento, e dunque invia la richiesta al *Language Server* che supporta il determinato linguaggio.



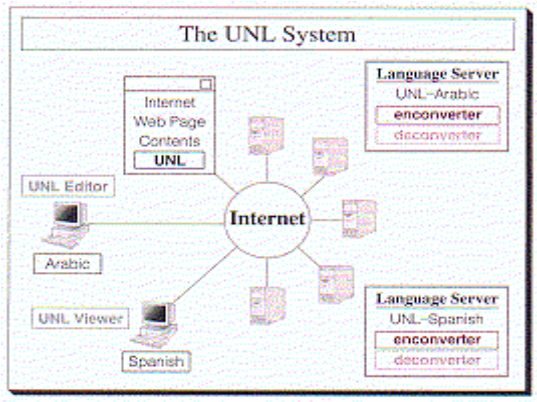
L'editor/viewer è connesso al Language Server: all'editor vien proposto un documento da tradurre

Il *language Server* esegue dunque la *encodifica* del documento in *UNL*.



L'editor riconosce la lingua del testo ed invia la richiesta al Language Serve esegua l'encodifica

Una volta che il documento è stato elaborato dal *software* di *EnCo*, esso può essere visualizzato come serie di espressioni *UNL*, magari per una redazione del documento in modo interattivo oppure, l'applicazione finale, tramite il *software* di *DeCo*, presente sui



Il Language Server esegue la decodifica del documento UNL e lo invia ai \ *Language Server*, riconvertito in lingue, come ad esempio lo Spagnolo, ed inviato ai *viewer* che visualizzeranno nel *browser* degli utenti connessi il documento redatto nella lingua finale.

5.6 Sperimentazioni con UNL

5.6.1 UNL e la traduzione automatica delle schede bibliografiche

Il sistema *UNL* è dunque l'applicazione più ambiziosa di *M.T.* per adesso implementata, ed essa si adatta alla resa *multilingue* di parti di testo specifiche, tecnico-scientifiche, e comunque il cui dizionario di parole risulti abbastanza settoriale: tutte quelle caratteristiche che ho riconosciuto come perfettamente proprie delle schede bibliografiche pubblicate online.

Infatti nella realizzazione delle pagine web ero riuscito a fornire un servizio di *C.L.I.R.*, ma secondo i criteri del *M.L.I.A.*, avrei dovuto ottimizzare il sistema di *Information Retrival* rendendo capace il maggior numero possibile di utenti di poter leggere i contenuti dei testi archiviati nella propria lingua, o comunque in un linguaggio a lui accessibile secondo i criteri di abbattimento dello *Standard Inglese* precedentemente accennati.

Ciascuna scheda bibliografica, non avrebbe dovuto essere tradotta integralmente, anche perché alcune informazioni, quali :

1. autore,
2. *titolo*,
3. *Soggetto topografico*,
4. *ISBN* o *ISSN*,
5. *data di edizione*,
6. *luogo di pubblicazione*

per una precisa indicazione del testo descritto, avrebbero dovuto rimanere nella lingua di redazione originale; la traduzione dei campi si sarebbe dunque limitata ai campi riferiti al:

1. *tipo di documento*,
2. *soggetto*,
3. *riassunto del testo*

e purtroppo, eccetto il campo *tipo documento*, i campi che in fieri avrebbero contenuto molto testo, e quindi avrebbe dovuto essere creata una nuova tabella nel database *bibliografiapiste* identica alla tabella *libri*, nella quale i contenuti

dei campi *Type*, *Subject* ed *Abstract*[115] fossero stati o direttamente rappresentati tramite espressioni *UNL*, oppure fossero state redatte nella lingua per la quale il processo di *encodifica* in *UNL* fosse il più preciso e sicuro per poi poter *decodificare* in tutte le lingue supportate dai *Server Language*.[116]

La scelta è ovviamente caduta sulla seconda opzione, e per contrappasso,[117] la lingua più malleabile per l'encodifica sarebbe stata l'*Inglese*.

La tabella *Book*, praticamente gemella della tabella *libri*, destinata ad archiviare la bibliografia adattata alle operazioni di *M.T.* tramite *UNL* avrebbe avuto la seguente architettura.

Tipo documento :	*UNL* o *Inglese*
ISBN / ISSN :	Codice
Autore :	Lingua originale
Titolo :	Lingua originale
Edito :	Lingua originale + nume
Soggetto :	*UNL* o *Inglese*
Soggetto Topografico :	Lingua originale
Riassunto :	*UNL* o *Inglese*

Malgrado *UNL* costituisse l'applicazione più efficiente, con cui fornire il supporto multilinguistico, e malgrado il corpus di testi limitato (rispetto agli standard dei corpora testuali), non sarebbe stato possibile estendere a tutta la bibliografia il servizio multilingua.

La sperimentazione avrebbe riguardato solo una parte del corpus bibliografico, ed avrebbe costituito un esempio funzionante applicato ad una porzione rappresentativa di testi che avessero potuto esemplificare e ben descrivere tutte le differenti tipologie di documenti presenti nel corpus.

Questo per dimostrare, a parte la peculiarità dell'argomento bibliografico trattato, che sarebbe stato possibile fornire questo servizio per tutto quel materiale documentario relativo a temi *Egittologici*, anche i più particolari, fra i quali sarebbe doveroso inserire la bibliografia relativa alle *Piste Carovaniere del Deserto Occidentale Egiziano*.

5.6.2 estendere UNL

Come introdotto precedentemente, la sperimentazione tramite *UNL* avrebbe riguardato un campione rappresentativo dei documenti presenti nel corpus di

[115] Quello che nei campi corrispondenti della tabella *libri* avevo scritto in *Italiano*.

[116] Per i processi di *encodifica* e *decodifica* è necessario rifarsi a quanto sviluppato nella 3° parte dedicata alle strutture di *Machine Translation*.

[117] Contrappasso rispetto alle premesse filosofiche tramite cui è stato propagandato il sistema *UNL*, ossia sopperire al dominio della lingua inglese nelle applicazioni web.

testi della bibliografia digitale, ma sarebbe auspicabile nell'ottica del futuro, poter prima di tutto estendere a tutto il corpus bibliografico questo supporto multilinguistico, ed estendere sempre più la loro traduzione a nuovi *Language Server*, dilatando sempre maggiormente la panoramica di linguaggi supportati.

Proprio in questa ottica, la sperimentazione realizzata sarebbe stata eseguita proprio per non limitare a pura teoria il tema del multilinguismo, e deve quindi essere presa come tale: un esempio che è possibile applicare ai *corpora bibliografici*, la cui qualità potrebbe essere espansa in breve tempo ad un enorme numero di utenti, il cui valore informatico sarebbe incrementato tramite un applicativo più che notevole, forse costituente il top di tutti i sistemi di *M.T.* mai concepiti, incrementando la comprensione e la facilità di studio dei contenuti trattati, a prescindere dalla lingua con cui essi furono originariamente redatti.

5.6.3 UNL applicato ad un sistema di C.L.I.R.

L'utilizzo di *UNL* come sistema di *M.T.* è soltanto uno, forse il più ambizioso, tra gli applicativi realizzabili.

All'interno dello stesso dominio della traduzione automatica è possibile rivolgersi al sistema *UNL* per applicazioni molto particolari, ed in questo specifico caso, avendo sviluppato all'interno del sistema di consultazione dell'archivio bibliografico il concetto e quindi un prototipo di sistema di *C.L.I.R.*, è stato ipotizzato l'utilizzo di *UNL* proprio all'interno di un sistema di *Retrival Multilinguistico*.

Questa ipotesi è nata durante l'analisi dei diversi processi che avrebbero portato e permesso ad un sistema automatico di rispondere correttamente ad una stringa complessa espressa in *N.L.*, estraendo i documenti pertinenti e prescindere per entrambi dai linguaggi di redazione.

La difficoltà principale, a parte la costruzione dell'*ontologia infor* per la rappresentazione astratta dei contenuti dei documenti della bibliografia, rimaneva la realizzazione di un sistema capace di indicare al motore di ricerca come corrette tutte quelle parole chiave rappresentative di ogni elemento dell'ontologia e la loro traduzione nei linguaggi supportati dal portale.[118]

Era necessario cioè implementare un sistema che comunicasse al motore di ricerca che, ad esempio, la chiave keyword(x) della tabella *infor* espressa nella lingua x, all'interno del ricerca nata dalla richiesta della stringa di *query* così rappresentabile *query* = [keyword1(x), keyword2(x), keyword3(x), ecc…], era equivalente alla chiave espressa nelle lingue y, z, ecc.., e dunque avrebbero prodotto l'estrazione dello stesso elemento dalla tabella *Infor*.

In pratica l'effetto prodotto dall'ontologia *infor* e dal sistema di ricerca così concepiti, è quello di simulare rispondendo alla richiesta espressa in lingua

[118] Naturalmente il tentativo sarebbe stato quello di rendere il numero di linguaggi supportati il più esteso possibile.

x come se essa fosse stata contemporaneamente eseguita, *con lo stesso significato*, espresso nelle lingue x, y, z, ecc…; infatti è addirittura possibile formulare query con parole contemporaneamente in più lingue come nell'esempio: "*caravan's tracks del deserto occidentale*", interrogazione improbabile, ma che illustra a dovere le potenzialità del prototipo del sistema di *C.L.I.R.*.

Il sistema simula inoltre l'effetto di portare *query* d'interrogazione ed i campi dell'ontologia allo stesso standard rappresentativo (la stessa lingua), perché in pratica sembra formulare interrogazioni in qualsiasi dei linguaggi supportati ed ottenere una corretta interrogazione ed estrazione degli argomenti rappresentati nell'ontologia, base della successiva interrogazione per l'estrazione dei testi pertinenti.

L'applicazione di *UNL* a questo triplice sistema *query* **à** *ontologia* **à** *archivio testi* invece di simulare questo processo, porterebbe all'effettiva riduzione delle lingue in gioco precedentemente descritta.

UNL non verrebbe utilizzato però nel suo modo tradizionale, ossia:

1. partendo da un testo sorgente,
2. encodificandolo in espressione *UNL*,
3. decodificando per produrre i testi target

poiché per l'applicazione in questione non sarebbe stato necessario tradurre niente, ma riuscire ad astrarre le chiavi di ricerca e la query immessa dall'utente allo stesso codice astratto ed indipendente dai diversi linguaggi sorgenti in gioco.

Il linguaggio *target* non sarebbe dunque più consisistito in un *N.L.*, ma sarebbe stato *UNL* stesso, e il suo utilizzo *UNL* sarebbe da arresare al punto 2, evitando inoltre proprio il processo più difficoltoso da realizzare, la decodifica da *UNL* a linguaggi *target*.

Il processo di *encodifica* avrebbe riguardato due parti altrettanto distinte del processo di *I.R.* e sarebbe avvenuto in due momenti distinti:

1. *Query d'interrogazione*: la stringa espressa in *NLsoruce*(x) dovrebbe essere encovertirla ed il processo, ovviamente, sarebbe avvenuto in tempo reale.
2. *Ontologia Infor*: l'ontologia non avrebbe più necessitato della traduzione delle chiavi relative a ciascun elemento in tutte le lingue supportate, poiché ciascun elemento avrebbe avuto come proprie le stesse chiavi espresse in *UNL*.
   Praticamente ogni campo "*chiavi*" avrebbe dovuto essere scritto in una sola lingua, meglio se l'*Inglese*, e

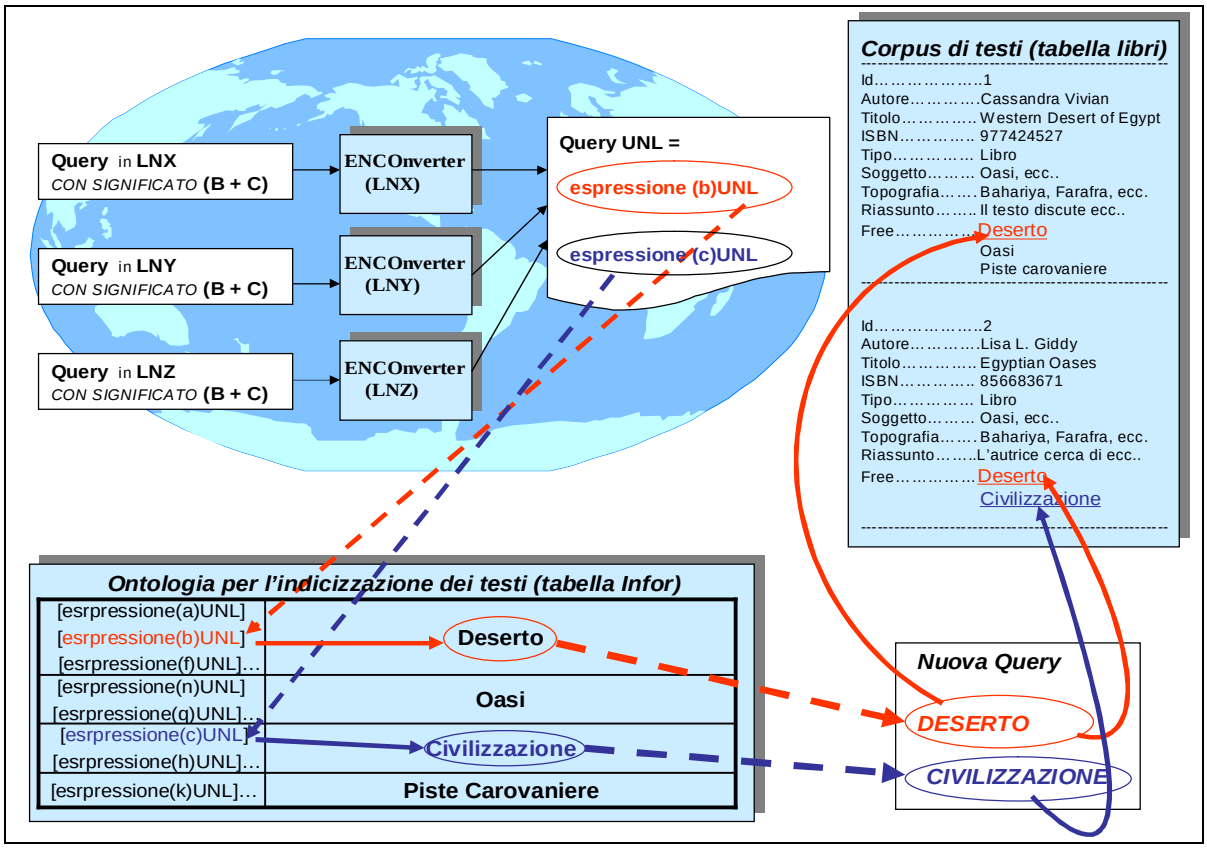
poi sottoposta ad un software *ENCO*, le cui espressioni risultanti avrebbero costituito le chiavi *UNL* riferite a quel determinato argomento.
Questo processo, evidentemente, dovrebbe avvenire prima di caricare nel database Infor i valori delle singole chiavi, anche se non è da escludere che pure questo procedimento potrebbe avvenire in tempo reale durante l'interrogazione dell'utente.

L'encodifica produrrebbe quindi la riduzione delle diverse lingue in gioco (fra *query* e *chiavi*) allo stesso standard linguistico: *UNL*.
Il sistema in concreto potrebbe funzionare così:

- Previa riduzione dei campi chiavi della tabella infor ad espressioni *UNL* o in tempo reale;

- Ricezione della *query* e sua encodifica ;

- La *query*(*UNL*), a questo punto, potrà essere utilizzata per interrogare l'ontologia ed estrarne i campi pertinenti.



La realizzazione di questo progetto assicurerebbe un netto miglioramento dell'efficienza di tutto il sistema di *I.R.* poiché produrrebbe:

1. *la riduzione delle dimensioni dell'ontologia Infor*, eliminando la traduzione, linguaper lingua, delle chiavi di ciascun argomento ivi rappresentato;

2. *il mantenimento delle sfumature semantiche* contenute nell'espressione di *query* ed all'interno delle chiavi dell'ontologia *Infor*, ottenibile grazie alle caratteristiche intrinseche ad *UNL* stesso, ed a sua volta riuscire ad ottenere:

   2.1 *l'incremento della precision*:[119] il mantenimento del contenuto semantico delle espressioni di *query* diverrebbe un ulteriore criterio per l'esclusione degli elementi, *Ret & Nrel*, e cioè non pertinenti al "senso" dell'interrogazione, come ad esempio gli *omografi*;

[119] Per quanto concerne il concetto della Precision e di elementi Nrel ed elementi Ret vedi 2.3 *Sistemi di Information Retrival*

2.2 *la riduzione del processo di normalizzazione e di stemming*: assumendo quanto affermato nel punto 2 e 2.1 appare evidente che la riduzione a *stem* di una parola flessa potrebbe portare alla perdita delle sfumature semantiche necessarie a distinguere il senso di una *query* da altre.
Il livello di normalizzazione e stemming dovrebbe dunque essere ponderata, perché invece di avere come risultato l'incremento dell'efficienza provvedendo alla eliminazione di elementi ridondanti ed inutili ne produrrebbe un decremento perchè cancellerebbe il significato semantico contenuto, ad esempio, in una particolare costruzione sintattica.

2.3 *Riduzione complessiva dei tempi*: i tempi di encodifica della stringa di *query* [120] andrebbero a sostituirsi ai tempi di *normalizzazione* e di *stemming* riducendo sicuramente il tempo totale necessario alla rielaborazione della *query* dal testo sorgente alla stringa che sarà utilizzata per l'interrogazione.

Sembra dunque ragionevole ipotizzare che l'applicazione di *UNL* al prototipo di sistema di *C.L.I.R.* sperimentato nel *portale egittologico Bibliografiapiste* potrebbe produrre risultati molto interessanti, o comunque confrontabili con i sistemi standard.

[120] Che per quanto complessa possa mai essere, sarà sempre una frase pensata per un'interrogazione ad un archivio.

Bibliografia

Aiello Mario, Albano Antonio, Attardi Giuseppe, Montanari Ugo, *Teoria della computabilità, logica, teoria dei linguaggi formali*, ETS Editrice, Pisa, I ristampa, 1979.

Balossino Nello, *Informatica: dall'informazione di base alla grafica computerizzata, terza edizione ampliata*, S. Lattes & C. editori, Torino, 1989.

Blanc Frédéric, Brandeis Pierre, *Lavorando in Turbo Pascal: manuale – note didattiche – esempi*, traduzione ed adattamento dell'originale in lingua Francese *Turbo Pascal sur IBM/PC*, a cura di Bruno Marchisio e Laura Ombra, Petrini editore, Torino, 1992.

Cassel P., *MySQL & mSQL*, traduzione dell'originale in lingua Inglese *MySQL & mSQL database for moderate-sized organizations & web sites*, HOPS ed., 2000.

Choi W., Kent A., Lea C., Prasad G. e Ullman C., *PHP 4 Guida per lo sviluppatore*, traduzione dell'originale in lingua inglese *PHP 4*, Hoelpi ed., 2001.

Ciaccia Paolo, *Information Retrival, Dispense del Corso di Sistemi Informativi II, a. a. 1999 – 2000*, DEIS – CSITE – CNR, Università di Bologna, 2000.

De Francesco Nicoletta, De Nicola Rocco, *Università degli Studi di Pisa – Dipartimento di Informatica: Semantica Operazionale e Denotazionale di un Semplice Linguaggio Funzionale*, Servizio Editoriale Universitario di Pisa (SEU), 1990.

Frosini Graziano, *Il Linguaggio di Programmazione Pascal ed il suo utilizzo su Personal Computer*, ETS Editrice, Pisa, II edizione 1989.

Ghierchia Gennaro, *Le strutture del Linguaggio Semantica*, Il Mulino Editrice, Bologna, 1997.

Goodman Danny, *JavaScript la guida - quarta edizione*, traduzione dell'originale in lingua inglese *JavaScript bible, fourth edition*, Mc Graw Hill ed., 2001.

Graffi Giorgio, *Le strutture del Linguaggio Sintassi*, Il Mulino Editrice, Bologna, 1994.

Greenspan Jay, Bulger Brad, *Sviluppare applicazioni per database con MySQL/PHP*, traduzione dell'originale in lingua inglese *My SQL/PHP Database applications*, Apogeo ed., 2001.

Hearst Marti A., *The Use of Categories and Clusters for Organizing Retrival Results*, in Strzalkowski T., *Natural Language Information Retrival*, pp. 334 – 349, Kluwer Academic Publisher, 1999.

Holzschlag Molly E. et al., *XML, HTML, XHTML Magic*, traduzione dell'originale in lingua Inglese *XML, HTML, XHTML Magic*, Addison Wesley ed., 2002.

Jensen Kathleen, Wirth Niklaus, *Pascal: manuale e standard del linguaggio*, Gruppo Editoriale Jackson, Milano, 1981.

Katzner Kenneth, *The Languages of the World Revised Edition*, Routledge & Kegan Paul LTD, London & New York, 1998.

King Robert, *Linguistica Storica e Grammatica Generativa*, Il Mulino Editrice, Bologna, 1973.

Kline Kevin, Kline Daniel, *SQL Guida di riferimento*, traduzione dell'originale in lingua inglese *SQL in a nutshell*, Apogeo ed., 2001.

Lazzerini B., Frosoni G., *Università degli Studi di Pisa - Facoltà di Ingegneria - Istituto di Elettronica e Telecomunicazioni: Introduzione alle strutture dati (utilizzando il linguaggio Pascal)*, Servizio Editoriale Universitario di Pisa, 1989.

Lerdorf Rasmus, Tatroe Kevin, *Programmare in PHP*, traduzione dell'originale in lingua Inglese *Programming PHP*, O'Reilly, Hops edizioni, 2002.

Lerdorf Rasmus, Tatroe Kevin, *Programmare in PHP*, traduzione dell'originale in lingua Inglese *Programming PHP*, O'Reilly, Hops edizioni, 2002.

Meloni Julie, *PHP per esempi*, traduzione dell'originale in lingua Inglese a cura di Prima Communications, Inc., *PHP – Fast & Easy web Development*, Gruppo Editoriale Futura, 2001.

Mohammed J. Kabir, *Manuale Apache Server - Guida per l'amministratore*, traduzione dell'originale in lingua inglese *Apache Server Administrator's handbook*, Jackson ed., 2001.

Parisi Domenico, Castelfranchi Cristiano, *La Macchina e il Linguaggio*, Bollati Boringhieri Editore, Torino, 1987.

Peterson Richard, *Linux: la Guida Completa, Terza edizione*, traduzione dell'originale in lingua Inglese *Linux: The Complete Reference, Fourth Edition*, McGraw-Hill Companies, 2002.

Powell A. Thomas, *HTML la Reference*, traduzione dell'originale in lingua inglese *HTML: The complete reference, third edition*, Mc Graw Hill, 2001.

Riloff Ellen, Lorenzen Jeffrey, *Extraction-Based Text Categorization: Generating Domain-Specific Role Relationships Automatically*, in Strzalkowski T., *Natural Language Information Retrival*, pp. 167 - 196, Kluwer Academic Publisher, 1999.

Uchida Hiroshi, Zhu Meiying, Della Senta Tarcisio, *The UNL, a Gift for a Millennium*, Institute of Advanced Studies - The United Nations University, 1999.

Scalise Sergio, *Le strutture del Linguaggio Morfologia*, Il Mulino Editrice, Bologna, 1994.
Simi Maria, *Sistemi per l'archiviazione e recupero delle informazioni*, Dipartimento di Informatica Università di Pisa, a.a. 1996 – 1997.

Smeaton F. Alan, *Using NLP or NLP Resources For Information Retrival Tasks*, in Strzalkowski T., *Natural Language Information Retrival*, pp. 99 – 111, Kluwer Academic Publisher, 1999.

Sparck Jones, Karen, *What is the Role of NLP in Text Retrival*, in Strzalkowski T., *Natural Language Information Retrival*, pp. 1 – 24, Kluwer Academic Publisher, 1999.

Stucky Matthew, *MySQL*, traduzione dell'originale in lingua Inglese *MySQL: building user interfaces*, Mc Graw Hill ed., 2001.

Tobias Ratschiller, Till Gerken, *PHP 4.0 applicazioni web*, traduzione dell'originale in lingua inglese *web application development with PHP 4.0*, Addison Wesley ed., 2001.

Ullman Larry, *PHP per il World Wide Web*, traduzione dell'originale in lingua inglese *PHP for the World Wide Wide Visual quick start guide*, Addison Wesley ed., 2001.