Università degli Studi di Udine

**Dipartimento di Matematica e Informatica**

Via delle Scienze, 206 - 33100 UDINE (ITALY)
tel. (39) (432) 558400 fax (39) (432) 558499

UDMI/ 13/ 96/ RR

# Sviluppi futuri di FIRE

**Stefano Mizzaro**

Abstract

In questo documento espongo alcune idee sugli sviluppi futuri del sistema FIRE allo scopo di gettare le basi per il progetto delle prossime versioni del sistema. Gli argomenti trattati sono piuttosto eterogenei ma, come si vedrà, vi sono alcuni punti in comune che giustificano una trattazione unica.

Rapporto interno a circolazione limitata

Internal report; limited circulation only

Sono stati assolti gli obblighi di legge (D.L. Lgt. 31/8/1945 n. 660)

# Sviluppi futuri di FIRE

**Stefano Mizzaro**

Dipartimento di Matematica e Informatica
Università di Udine
V. delle Scienze, 206 – Loc. Rizzi – 33100
Udine – Italy
e-mail: mizzaro@dimi.uniud.it

Febbraio 1995

In questo documento espongo alcune idee sugli sviluppi futuri del sistema FIRE allo scopo di gettare le basi per il progetto delle prossime versioni del sistema. Gli argomenti trattati sono piuttosto eterogenei ma, come si vedrà, vi sono alcuni punti in comune che giustificano una trattazione unica.

## 1. Introduzione

In questo documento espongo alcune idee sugli sviluppi futuri del sistema FIRE, descritto in Floreanini (1993), Mizzaro (1995) e Brajnik et al. (1990a, 1990b, 1991a e 1991b), allo scopo di gettare le basi per il progetto delle prossime versioni del sistema. Le considerazioni qui esposte non hanno pretese di completezza e correttezza, ma sono piuttosto spunti da tenere in considerazione in fase di progetto di una nuova versione del sistema.

I possibili sviluppi futuri del sistema FIRE si diramano in varie direzioni, in buona parte indipendenti fra di loro: dopo aver brevemente illustrato la situazione di FIRE nella sezione 2, nelle seguenti sezioni descrivo ognuno di tali sviluppi. Vi sono però collegamenti fra tali sviluppi, illustrati nella sezione conclusiva. Più in dettaglio, questo documento è strutturato nel modo seguente.

Nella sezione 2 richiamo brevemente le linee principali del sistema FIRE, allo scopo di introdurre la terminologia necessaria nel seguito.

Nella sezione 3 presento un possibile miglioramento del thesauro: illustro come sia possibile introdurre nuovi tipi di archi e contemporaneamente mantenere una completa compatibilità con lo standard comunemente accettato che prevede unicamente le relazioni bt, nt, rt e ut.

Nella sezione 4 descrivo un nuovo approccio alla riformulazione tramite navigazione sul thesauro e tecniche morfologiche. Tale nuovo approccio dovrebbe comportare un sensibile miglioramento per quanto riguarda il numero e la pertinenza dei termini suggeriti all'utente durante la fase di riformulazione.

Nella sezione 5 affronto il problema della descrizione del bisogno informativo tramite un enunciato in linguaggio naturale e della traduzione di tale bisogno informativo in un linguaggio formale comprensibile dal sistema.

La sezione 6 descrive a grandi linee le caratteristiche della nuova interfaccia utente di FIRE, necessaria a causa delle limitazioni dell'interfaccia attuale.

Nella sezione 7 accenno ad altri vari miglioramenti auspicabili per il sistema. Fra gli altri, affronto i problemi dell'impossibilità di esprimere, utilizzando un sistema booleano o vector space, relazioni fra concetti (ad esempio relazioni di causa-effetto), della mancanza di *spiegazioni* (il sistema non fornisce all'utente spiegazioni sul suo comportamento) e del mancato utilizzo di altre informazioni ricavabili dall'utente (ad esempio che cosa l'utente desidera fare con i documenti reperiti).

Anche se è vero che le varie migliorie a FIRE possono essere apportate in modo indipendente, vi sono comunque delle importanti relazioni fra di esse: nella sezione 8 presento i legami fra gli sviluppi illustrati nelle sezioni precedenti. La sezione 9 conclude questo lavoro.

## 2. Il sistema FIRE

Non fornirò in questa sede una descrizione completa di FIRE; mi limiterò ad accennarne le linee principali, rimandando per ulteriori particolari alle referenze citate in precedenza.

Un *Sistema di Information Retrieval Intelligente* (IIRS), è un *Sistema di Information Retrieval* (IRS) nella cui implementazione si sono sfruttate tecniche di *Intelligenza Artificiale* (IA) allo scopo di consentire una migliore formulazione del bisogno informativo dell'utente. Le tecniche d'IA sembrano indispensabili per conseguire risultati ottimali, come affermato ad esempio in Croft (1993) o Ingwersen (1992).

Il sistema FIRE (acronimo di *Flexible Information Retrieval Environment*) è un IIRS. Esso rientra nel più ampio ambito del Progetto FIRE, orientato allo studio, allo sviluppo ed alla sperimentazione di un ambiente flessibile per la costruzione di *interfacce intelligenti* per sistemi per l'interrogazione di banche dati bibliografiche. Più in dettaglio, tale progetto ha come obiettivo la realizzazione di un sistema (il sistema FIRE) che simuli alcune capacità tipiche di intermediari (di un sistema tradizionale di *information retrieval*) e bibliotecari (di una biblioteca), in tal modo facilitando la ricerca di dati bibliografici.

L'obiettivo di rendere il più possibile automatico il supporto all'utente nella ricerca di informazioni bibliografiche viene perseguito con l'utilizzo di tecniche d'IA. Infatti, componenti fondamentali di FIRE sono le seguenti 2 *basi di conoscenza*: conoscenza di dominio (realizzata tramite il thesauro) e conoscenza esperta (implementata in parte dichiarativamente con regole di produzione e in parte proceduralmente) e conoscenze relative ai vari utenti che accedono al sistema (modellazione dell'utente).

Scopo principale di FIRE è quindi aiutare l'utente ad individuare ed esprimere con precisione il proprio bisogno informativo, impiegando la terminologia più appropriata in relazione al contenuto ed alla organizzazione della banca dati. La formulazione del problema informativo

viene poi automaticamente tradotta nel linguaggio di interrogazione interno del sistema, il quale gestisce l'accesso alla banca dati in maniera completamente trasparente all'utente.

Fra le funzionalità di FIRE, la caratteristica peculiare è il supporto fornito durante la fase di *riformulazione* della query: il sistema propone all'utente alcuni termini sinonimi dei (o correlati ai) termini introdotti in precedenza, allo scopo di meglio definire il bisogno informativo. I termini suggeriti dal sistema, se accettati dall'utente, vengono inclusi nella query. La scelta dei termini da proporre viene effettuata da un *sistema esperto*, che utilizza le basi di conoscenza del sistema (per ulteriori dettagli si veda Mizzaro, 1995).

La riformulazione attuale presenta dei problemi: vengono proposti troppi termini all'utente, egli rifiuta la maggior parte di essi, l'ordine in cui i termini vengono proposti non corrisponde alla probabilità che essi vengano accettati e alcuni termini vengono riproposti, talvolta senza motivo. Questi problemi dipendono secondo me da due fattori: il sistema esperto non sfrutta tutte le conoscenze disponibili nelle basi di conoscenza del sistema e tali basi di conoscenza non contengono abbastanza informazioni. Le sezioni 3 e 4 contengono alcune proposte per risolvere tali problemi.

In figura 1 è illustrata l'architettura del sistema FIRE. Vi compaiono le seguenti componenti:

- il modulo UI (User Interface), realizzato secondo il paradigma WIMP (Window Icon Menu Pointer). L'interfaccia può essere scomposta in due parti: IRESFACE, che costituisce il principale canale di comunicazione con l'utente e DSKBED, descritto fra poco;
- il modulo IRES (*Information Retrieval Expert Subsystem*), incaricato della simulazione dell'intermediario. Contiene la base di conoscenza EKB (*Expert Knowledge Base*), che codifica sotto forma di regole di produzione le conoscenze di un intermediario;
- il modulo DSKBMAN (per DSKB Manager), incaricato dello sviluppo della DSKB (*Domain Specific Knowledge Base*, ossia thesauro utilizzato come base di conoscenza). L'utente può accedere ai servizi del DSKBMAN tramite l'interfaccia DSKBED (DSKB Editor) che permette di esaminare e modificare il thesauro;
- il modulo DBMAN (*Data Base Manager*), gestore delle banche dati a cui il sistema accede;

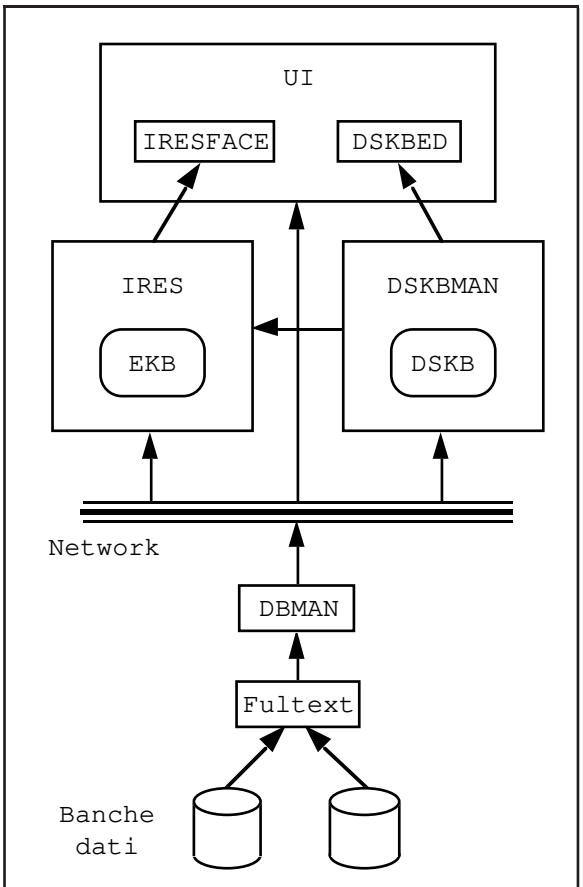

Figura 1. Architettura del sistema FIRE.

- l'IRS booleano FulText commercializzato da Fulcrum S.p.A.;
- le 2 banche dati bibliografiche. Per ora FIRE può accedere a 2 banche dati (o *collezioni*). La prima è costituita da circa 2000 documenti ricavati dalla banca dati INSPEC e relativi alle applicazioni di tecniche di intelligenza artificiale nell'industria. La seconda è denominata BSF (acronimo di Bibliografia Storica Friulana) ed è costituita da circa 5000 documenti relativi ad aspetti storici, economici e legislativi della vita montana del Friuli-Venezia Giulia.

Le migliorie proposte in questo documento riguarderanno soprattutto il modulo IRES e, in misura minore, i moduli UI e DSKBMAN.

In questa sede non approfondisco ulteriormente la descrizione di FIRE; per ulteriori dettagli rimando alla bibliografia sopra citata. Nelle prossime sezioni espongo alcune considerazioni che, secondo me, guideranno gli sviluppi futuri del sistema.

## 3. Thesauro stratificato

In questa sezione presento un possibile miglioramento del thesauro: illustro come sia possibile introdurre nuovi tipi di archi e contemporaneamente mantenere una completa compatibilità con lo standard comunemente accettato che prevede unicamente le relazioni bt, nt, rt e ut.

Nella sezione 3.1 richiamo alcune nozioni di base dei thesauri, illustrando come un thesauro possa essere utilizzato come base di conoscenza di un IIRS durante il processo di riformulazione della query. Nella sezione 3.2 propongo un particolare tipo di rete semantica, utilizzato nella sezione 3.3 per mostrare come costruire un thesauro che ammetta archi differenti dai consueti bt, nt, rt e ut e contemporaneamente mantenga una completa compatibilità con lo standard comunemente accettato.

### 3.1. Introduzione

Un thesauro può essere considerato una rete semantica (si veda Rich, 1986): vi sono nodi (i termini) e relazioni fra di essi (nt, bt, rt, ut). Tale rete semantica implementa una *base di conoscenza*, in quanto contiene la conoscenza terminologica di un certo dominio e alcuni legami di tipo *semantico* fra termini.

Un thesauro è però un caso particolare di rete semantica: i nodi sono termini di un vocabolario e le relazioni sono *esclusivamente* di tipo nt, bt, rt, ut. L'utilizzo delle sole relazioni nt, bt, rt, ut è da una parte necessario per uniformarsi allo standard comunemente accettato, ma d'altra parte porta ad una perdita di conoscenza: nelle reti semantiche infatti si utilizzano solitamente numerose altre relazioni, quali isa, instance-of, has-part, part-of, causes, ecc. e quindi, relazioni che sarebbe più naturale classificare ad esempio di tipo causa-effetto o processo-prodotto, ecc. vengono di solito inglobate in relazioni rt, assieme a relazioni di altro tipo. Allo stesso modo, la ut viene usata per rappresentare sia relazioni di sinonimia vera e propria che relazioni molto più deboli, e la bt sia is-a che instance-of che part-of (per un elenco più completo, anche se non ancora definitivo, si veda Danesi, 1990, pp. 69-74).

L'esigenza di una rappresentazione della conoscenza più completa si evidenzia quando si vuole simulare il comportamento dell'intermediario (come è il caso del sistema FIRE) durante l'attività di *riformulazione* della query. Intuitivamente si può osservare che se la navigazione è eseguita da un intermediario umano, egli utilizza in realtà più informazioni di quelle che otterrebbe dal thesauro, ovvero informazioni che derivano dalle conoscenze proprie e dell'utente; se la navigazione è invece automatica (ovvero effettuata autonomamente dal sistema), con un thesauro di tipo standard il sistema non possiede le conoscenze dell'intermediario e quindi deve utilizzare pesantemente quelle dell'utente. Nel caso di FIRE, questo problema si concretizza, come già accennato in precedenza, nell'eccessivo numero di termini proposti da FIRE all'utente durante la fase di riformulazione del bisogno informativo. L'utilizzo di un thesauro arricchito con vari tipi di relazioni presenta d'altra parte il problema della non aderenza allo standard.

La soluzione ai problemi illustrati finora è l'estensione del thesauro in modo *compatibile con lo standard attuale*, ma che permetta l'utilizzo di relazioni di nuovo tipo. Prima di illustrare tale soluzione, nella prossima sezione presento in modo generale un particolare tipo di rete semantica; tale rete semantica sarà poi utilizzata per la definizione del nuovo tipo di thesauro.

### 3.2. Rete semantica stratificata

La struttura di una rete semantica è rappresentabile mediante un grafo: vi sono nodi (che rappresentano concetti) e archi (che rappresentano relazioni) di vario tipo fra tali nodi.

Quando il numero di nodi e archi diviene troppo grande, come si verifica nel caso di un thesauro, la rete semantica diviene solitamente troppo complessa (anche dal punto di vista della rappresentazione grafica) per essere compresa interamente. È però possibile effettuare un'*astrazione* sulla rete semantica, individuando solo oggetti e relazioni che interessano: ciò corrisponde ad isolare un sottografo della rete semantica completa. Questo procedimento può essere raffinato: si consideri una rete semantica che si possa suddividere in più livelli (in seguito denominata *rete semantica stratificata*), illustrata in figura 2.

In tale rete semantica sono presenti 4 nodi (indicati con 1, 2, 3 e 4) e alcune relazioni fra tali nodi: a(1,2), b(2,3), a1(1,2), b1(2,3), ecc. La particolarità di questa rete è che essa è suddivisa su due livelli distinti: sopra al piano che separa graficamente tali livelli compaiono unicamente le relazioni a, b e c (primo livello) sotto il piano compaiono a1, b1, c1 e c2 (secondo livello). Si può pensare che queste ultime quattro relazioni siano delle *specializzazioni* delle relazioni al livello superiore: l'elemento 1 è in relazione a1 con l'elemento 2, *e quindi* è anche in relazione a; similmente, i nodi 2 e 3 sono in relazione b1, *e quindi* anche in relazione b e così via.

Il caso della relazione fra i nodi 3 e 4 va analizzato più attentamente: la relazione c del primo livello viene scomposta in 2 relazioni al livello

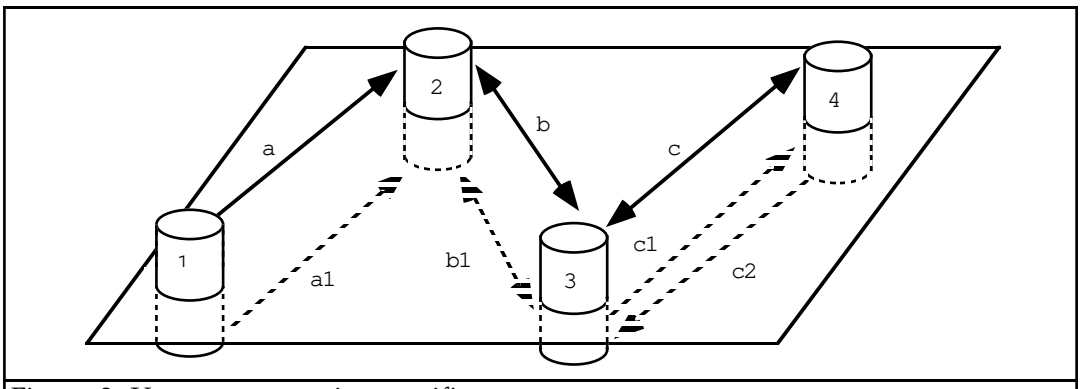


Figura 2. Una rete semantica stratificata.

sottostante, `c1` e `c2`.[1] Qui è da decidere se è sufficiente una sola delle relazioni `c1` o `c2` per concludere la relazione `c` o se sono necessarie entrambe.

In altri termini, si introduce una relazione di *gerarchia sulle relazioni*: indicando con `subrel` tale "metarelazione", si può dire che:

- `subrel(a1,a);`
- `subrel(b1,b);`
- `subrel([c1,c2],c)`, oppure `(subrel(c1,c) & subrel(c2,c))` a seconda dell'alternativa scelta[2].

Potrebbe essere utile anche assegnare ai nodi ruoli differenti a seconda del livello a cui ci si pone: ad esempio, un nodo al livello superiore potrebbe essere considerato un termine `t` e al livello inferiore un concetto `c` (il concetto rappresentato dal termine `t`); a questo punto, vi potrebbe essere una relazione di tipo `denota(t,c)`. Rimane però il problema di come rappresentare un concetto.

### 3.3. Il thesauro come rete semantica stratificata

Utilizzando una rete semantica stratificata a due livelli è possibile definire un nuovo tipo di thesauro. Il primo livello (il livello superiore di figura 2) è composto dalle relazioni `nt`, `bt`, `rt`, `ut`; tali relazioni possono poi essere particolareggiate nel livello sottostante, introducendo numerose sottorelazioni. Si ottiene in tal modo quello che si può denominare *thesauro stratificato*.

Nelle prossime sottosezioni indago sulle conseguenze che l'adozione di un thesauro stratificato può portare: dapprima illustro i vantaggi, poi affronto i problemi che si potrebbero presentare e infine propongo un possibile piano di lavoro per procedere in tale direzione.

---

1 Questo potrebbe essere ad esempio il caso, molto comune nei thesauri, di una relazione di tipo processo-agente che venga rappresentata al primo livello mediante un arco `rt` e al secondo livello mediante 2 relazioni, `process` ed `agent`. Tali relazioni potrebbero sussistere, ad esempio, fra i nodi `pastore` e `pascolare`.

2 Oppure si può pensare di introdurre un ulteriore tipo di metarelazione, senza limitarsi a `subrel`.

### 3.3.1. *Vantaggi*

Utilizzando un thesauro stratificato si ottiene un thesauro che contiene relazioni diverse dalle usuali `nt`, `bt`, `rt`, `ut` e che rispetta lo standard attuale (proprietà quest'ultima che non si otterrebbe se si estendesse il thesauro introducendo nuove relazioni in modo banale).

I vantaggi presentati dall'introduzione di nuove relazioni dovrebbero evidenziarsi durante la fase di riformulazione. Se in tale processo si utilizzano le *tattiche* (Bates, 1979), queste possono essere ridefinite, particolareggiando quelle classiche definite da Bates. In ogni caso, i termini proposti all'utente possono essere scelti in modo più accurato, grazie ad una maggiore quantità di informazioni.

Tali vantaggi dovrebbero risultare evidenti se l'utente può specificare nella query relazioni di vario tipo fra le faccette: in tal modo gli archi del thesauro corrispondenti a tali relazioni assumono una maggiore importanza rispetto agli altri. Ad esempio: se il bisogno informativo è di reperire documenti relativi alle relazioni di causa-effetto fra due concetti, gli archi `causes` del thesauro dovranno essere trattati in maniera particolare. Ma anche senza tali relazioni esplicite nella query l'utilizzo di un thesauro stratificato dovrebbe portare vantaggi: ad esempio, l'arco `rt` può essere utilizzato sia per relazioni di sinonimia che per la relazione di contrario, e questi due casi vanno indubbiamente trattati in modo differente.

Inoltre, secondo Gri (1993), non ci sono studi sull'aumento delle prestazioni che si avrebbe inserendo nuovi archi né sulle tattiche che potrebbero utilizzarli, quindi sarebbe interessante studiare le conseguenze di questo approccio. Se non si ottengono miglioramenti significativi, si può pensare che l'utilizzo di archi di vario tipo nel thesauro sia inutile, e provare ad utilizzare un thesauro con archi tutti dello stesso tipo (*correlazione generica* fra termini). Tale thesauro potrebbe essere generabile automaticamente dai documenti della collezione, per mezzo di un procedimento statistico basato sulla co-occorrenza di parole nei documenti (si veda ad esempio Salton, 1993). Dubito però che un thesauro con archi tutti dello stesso tipo fornisca prestazioni paragonabili ad uno con archi di tipo diverso: ad esempio, gli archi `bt` e `nt` possono essere usati per scegliere la faccetta in cui inserire un termine (si veda la sezione 4).

Si osservi che le quattro relazioni usuali posseggono le caratteristiche di generalità necessarie per poter essere particolareggiate in modo significativo: non si corre il rischio che in un thesauro standard non compaia una relazione fra due termini in qualche modo correlati, in quanto la `rt` può essere utilizzata per gli scopi più disparati.

È opinione abbastanza diffusa che sia praticamente impossibile riuscire a dare una descrizione corretta e completa tramite una rete semantica di un mondo realistico. Non è però di ciò che si ha bisogno: un thesauro di un IRS *supportivo* non deve necessariamente essere corretto e completo (anche se queste sarebbero senza dubbio caratteristiche positive), in quanto è usato solo per *proporre* termini all'utente, il quale ha l'ultima parola.

Facendo un passo ulteriore, si può pensare ad un thesauro a 3 livelli, con un livello più grezzo di quelli illustrati qui in cui gli archi siano tutti dello stesso tipo. Questo livello potrebbe essere *generabile automaticamente* e si può pensare di estenderlo in modo semi-automatico con l'ausilio di uno

schema di classificazione. Infatti, con tale schema si possono individuare gli archi bt/nt; le relazioni che restano sono quindi di tipo rt o ut e possono essere disambiguate utilizzando la parte strutturata dei documenti in quanto la relazione di tipo ut può sussistere solo fra termini controllati e non controllati.[3]

Un'ulteriore conseguenza dell'introduzione di relazioni di nuovo tipo si ha sulla *specificità* (Brajnik et al., 1991a): essa può forse essere ridefinita sulla base dei cammini isa/isa$^{-1}$ o part-of/has-part anziché sui cammini nt/bt, utilizzati per la definizione attuale.

Infine, l'implementazione di una rete semantica stratificata non presenta particolari problemi aggiuntivi rispetto ad una consueta rete semantica; è sufficiente tener conto della gerarchia fra relazioni e della conseguente suddivisione in livelli della rete semantica.

### *3.3.2. Problemi*

Un problema è la scelta di quali relazioni adottare per specializzare le nt, bt, rt e ut. Tale scelta è sicuramente dipendente dal dominio: ad esempio, una relazione di tipo causes, sebbene necessaria nella maggior parte dei thesauri, potrebbe essere superflua, ad esempio, in un thesauro geografico.

Un punto da chiarire è se sia possibile in linea di principio classificare un qualsiasi tipo di relazione come appartenente ad un'unica delle quattro classi individuate dalle relazioni nt, bt, rt, ut. Se è così, si può pensare ad uno studio generale di classificazione delle varie relazioni e ad una successiva applicazione di tale studio a un particolare dominio (ovvero, individuazione delle relazioni utili per quel dominio). Altrimenti, la classificazione delle relazioni va effettuata per ogni particolare dominio.

Per la gestione di un thesauro stratificato è indubbiamente necessario uno strumento software. Non può essere il costruttore del thesauro ad incaricarsi di controllare la consistenza e la mancanza di ridondanze (riferendosi alla figura 2, se c'è la relazione a1, è inutile che ci sia anche la relazione a, o meglio la relazione a può essere memorizzata in modo implicito), in quanto la complessità di tale compito porterebbe inevitabilmente ad errori.

Il problema più grosso è probabilmente la *fattibilità*: già è complesso costruire un thesauro classico, e un thesauro di questo tipo lo è senz'altro ancora di più. Comunque, è mia opinione che tale problema sembri più complesso di quanto non sia in realtà: dopotutto, quando si costruisce un thesauro e si deve creare una relazione (ad esempio, rt) fra due termini, si ha bene in mente quale è la relazione "reale" (ad esempio, causes) e quindi la costruzione di un thesauro stratificato potrebbe costare poca fatica in più rispetto ad un thesauro standard.

Anche l'estensione di un thesauro preesistente è secondo me un compito più semplice di quanto non sembri a prima vista. Utilizzando uno strumento automatico che prenda in input le relazioni di un thesauro e

---

[3] Se questo procedimento (in realtà un po' sbrigativo) non dovesse essere applicabile, si può comunque pensare di utilizzare un thesauro generato automaticamente con tutti gli archi dello stesso tipo e di estenderlo (particolareggiando ogni arco in nt, bt, rt, ut) con l'ausilio di uno strumento automatico analogo a quello menzionato alla fine della sezione 3.3.2.

chieda all'utente di particolareggiarle scegliendo relazioni da un menu, quanto tempo (in giorni/uomo) serve? Una stima *approssimativa* su un thesauro di 10.000 termini potrebbe essere la seguente:

- nel caso migliore si supponga di avere solo tre relazioni per ogni termine; con 10.000 termini si hanno quindi 30.000 relazioni. Se per particolareggiare una relazione sono necessari 30 secondi, allora servono 15.000 minuti, ovvero 250 ore, ovvero (8 ore al giorno) 30 giorni/uomo per processare tutto il thesauro;
- per stimare il caso peggiore, si pensi ad thesauro con 10 relazioni per ogni termine; con 10.000 termini si hanno quindi 100.000 relazioni. Quindi, con le stesse ipotesi del caso precedente, sono necessari 50.000 minuti, ovvero 100 giorni/uomo.

Tale stima di 30–100 giorni/uomo per un thesauro di 10.000 termini è puramente indicativa e inoltre vi è il problema di estendere il thesauro in modo, almeno in linea di massima, consistente, corretto e completo. A questo proposito, si ricordi comunque l'osservazione della sezione 3.3.1: in un sistema supportivo è in ogni caso l'utente ad avere l'ultima parola, quindi la proposta di un termine errato da parte del sistema non è un errore irrecuperabile.

### *3.3.3. Come procedere?*

Come si vede la situazione è piuttosto nebulosa: per comprendere se l'adozione di un thesauro stratificato porti ad effettivi vantaggi, e se sia una strada fattibile, propongo di effettuare quattro passi.

Il primo passo da compiere dovrebbe essere l'individuazione delle relazioni (fra cui potrebbero comparire ad esempio `synonym`, `contrary`, `syntactic-variant`, `causes`, `has-part`, `part-of`, `agent`, `recipient`, `verb-noun`, ecc., si veda Danesi, 1990, pp. 69-74) con cui particolareggiare le `nt`, `bt`, `rt`, `ut`. Questa scelta può essere effettuata, almeno in prima battuta, su uno specifico thesauro: si perde di generalità, ma si acquista in concretezza e semplicità.

Il secondo passo è poi quello di trovare esempi di riformulazione su piccoli sottoinsiemi del thesauro esteso che indichino che la strada sembra promettente. Questo lavoro dovrebbe essere effettuato simulando "a mano" i comportamenti di un sistema che navighi dapprima in un thesauro tradizionale e poi in un thesauro stratificato (ottenuto particolareggiando le relazioni di quello tradizionale) e confrontando tali comportamenti.

Questi confronti potrebbero anche indicare quali fra le relazioni introdotte hanno apportato i miglioramenti più sensibili e sono quindi da mantenere e quali non hanno invece modificato la situazione e possono quindi essere eliminate.

Se i due passi precedenti avranno prodotto risultati incoraggianti, bisognerà pensare ad implementare un sistema basato su un thesauro stratificato. Per fare ciò sono necessarie due attività: la realizzazione di un thesauro stratificato (preferibilmente a partire da un thesauro standard) e l'implementazione di nuove tattiche (o anche dell'intera base di conoscenza

esperta) in modo da sfruttare la conoscenza aggiuntiva fornita dal thesauro stratificato.

Infine (quarto passo), tale sistema andrà poi confrontato con il sistema basato sul thesauro standard, al fine di comprendere quali effettivi miglioramenti sono stati apportati.

## 4. Nuove strategie per la riformulazione

Ho già accennato in precedenza ai problemi presentati da FIRE durante la fase di riformulazione, quali il numero eccessivo di termini non rilevanti proposti all'utente, l'ordine in cui i termini vengono proposti, ecc. Uno degli obiettivi delle prossime versioni di FIRE dovrà senz'altro essere una riformulazione in cui pochi termini vengono proposti all'utente ed un'alta percentuale di essi vengono accettati.

Nella sezione precedente ho illustrato un possibile approccio per avvicinarsi a tale obiettivo: utilizzare una conoscenza terminologica del dominio più fine e completa di quella fornita da un thesauro di tipo standard, sfruttando un thesauro stratificato. In questa sezione propongo una strategia differente: migliorare il processo di riformulazione, ovvero le tecniche con cui i termini da proporre all'utente vengono scelti.

Nella prossima sottosezione descrivo al livello di dettaglio appropriato il processo di riformulazione in FIRE; nella sezione 4.2 evidenzio le lacune che tale processo presenta e nella sezione 4.3 propongo un approccio alla riformulazione alternativo che colmi tali lacune.

### 4.1. La riformulazione attuale

La *riformulazione* è l'attività svolta dal sistema al fine di meglio definire il bisogno informativo (b. i.) espresso in precedenza dall'utente. Tale attività consiste principalmente nella proposta all'utente di termini da parte del sistema. I termini suggeriti dal sistema, se accettati dall'utente, vengono inclusi nella query. In realtà il sistema può anche richiedere all'utente la conferma dell'eliminazione di termini (se si vuole diminuire il numero di documenti reperiti), ma qui non considero tale caso. Infatti, esso va probabilmente trattato in modo differente dall'aggiunta di termini e la sperimentazione effettuata finora su FIRE non ha dato chiare indicazioni su come procedere: a causa del ridotto numero di documenti nelle collezioni a cui FIRE può accedere, capita molto raramente che una query reperisca troppi documenti.

Nel sistema FIRE, come accennato nella sezione 2, vi è un modulo dedicato a tale attività, il modulo IRES. Nella versione attuale di IRES, la riformulazione viene effettuata in tre fasi:

(i) individuazione del *focus*: utilizzando varie informazioni (posting count e grado di interesse di termini e faccette, flag controllato/non controllato di termini, ecc.), viene individuato un termine su cui lavorare, detto focus;

(ii) a partire dal focus, si naviga sul thesauro[4], e si propongono all'utente i termini in relazione col focus. La navigazione è guidata dal tipo di arco che lega il focus con gli altri termini sul thesauro: si percorrono nell'ordine gli archi `rt`, `ms`, `tr`, `sb`[5], `bt` e `nt`;
(iii) i termini confermati dall'utente vengono aggiunti nella query *nella stessa faccetta* del focus.

Vi sono due critiche da muovere a tale approccio alla riformulazione: il focus viene scelto unicamente in base alla query e non utilizzando le informazioni ricavabili dal thesauro né quelle ricavabili dai documenti. Inoltre, la riformulazione viene effettuata a partire da *un unico termine* (appunto, il focus), e quindi durante la navigazione sul thesauro non sono tenuti in conto gli altri termini della query.

Tali critiche si rivelano fondate, come dimostrato dai problemi che l'approccio alla riformulazione presenta, analizzati nella prossima sezione.

### 4.2. Problemi presentati dalla riformulazione attuale

I problemi della riformulazione illustrata nella sezione precedente possono essere divisi in due gruppi, il primo specifico della fase di riformulazione e il secondo di carattere più generale, in quanto ogni IRS booleano ne è interessato. Tali problemi hanno comunque dei punti in comune e le soluzioni, come si vedrà nella sezione 4.3, sono correlate.

I problemi del primo gruppo (a cui in seguito mi riferirò come *problemi della riformulazione*) sono:

(i) il numero di termini proposti all'utente risulta spesso troppo elevato rispetto al numero di termini confermati;
(ii) l'ordine in cui i termini sono proposti è spesso in disaccordo con l'effettiva rilevanza dei termini (i termini proposti per primi non sempre sono quelli confermati dall'utente);
(iii) spesso vengono riproposti termini già proposti in precedenza, in quanto due termini possono essere in più relazioni o un termine del thesauro può essere in relazione con più termini della query. Si osservi che non sempre riproporre un termine all'utente è scorretto: talvolta uno stesso termine deve essere inserito in faccette diverse (come accade nel seguente esempio 2);
(iv) i termini vengono suggeriti in modo frammentato, in vari passi successivi; ciò potrebbe essere positivo se la rilevanza dei primi termini fosse superiore ma, essendo i termini ordinati in base alle

---

4 In realtà si utilizzano anche tecniche morfologiche: qui suppongo per semplicità che il thesauro contenga un arco *implicito* `ms` (per *morphological similitude*) che lega due termini se questi sono simili morfologicamente. Inoltre, `IRES` propone anche *troncamenti* di termini: qui suppongo che il thesauro contenga un arco *implicito* `tr` (per *troncamento*) che lega un termine al suo troncato (che suppongo appartenga al thesauro).

5 L'arco (implicito) `sb` è ottenuto prendendo i fratelli del termine di partenza nella gerarchia `bt`/`nt`, ovvero percorrendo consecutivamente gli archi `bt` e `nt`. Tale arco non è esplicito nel thesauro, ma, sempre per semplicità di esposizione suppongo che esista, come fatto per l'arco `ms`.

relazioni in cui sono con il focus, ciò non avviene (si veda il punto (ii)).

Vediamo ora il secondo gruppo di problemi, che denominerò *problemi dei sistemi booleani*. I due problemi di questo gruppo riguardano sempre la fase di riformulazione, ma sono di carattere più generale in quanto presenti in tutti i sistemi booleani. Il primo problema è che, quando il sistema propone un termine, l'utente decide se accettarlo o meno in base al suo b. i., e tende a non riflettere sul fatto che il sistema potrebbe inserire tale termine nella faccetta errata. Un esempio di tale situazione è il seguente.

*Esempio 1*

Si consideri il b. i.: "Reperire $x$ documenti che trattino il problema della produzione di carne". La query iniziale potrebbe essere costituita da 2 faccette, una contenente il termine 'produzione' e l'altra il termine 'carne'. Se tale query reperisce pochi documenti (meno di $x$), IRES inizia il processo di riformulazione, che potrebbe procedere nel modo indicato qui di seguito (con & indico l'*and* logico, con la virgola l'*or* logico, le parentesi tonde racchiudono i termini di una faccetta e con --(rif)--> indico un passo di riformulazione):

```
query iniziale: (Produzione) & (Carne)
--(rif)-->
(Produzione) &
(Produzione di carne,Carne)
--(rif)-->
(Produzione di carne,Produzione ) &
(Produzione di carne,Carne)
--(rif)-->
(Produzione di carne,Produzione,Carne) &
(Produzione di carne,Carne,Produzione)
```

Il problema è che il termine Produzione di carne andrebbe inserito, rispettando la concettualizzazione iniziale, nella faccetta con Produzione, ma il sistema lo propone prima nell'altra faccetta (si osservi che Produzione di carne è in relazione sia con Produzione che con Carne).

In casi di questo tipo, è lasciato all'utente il compito di introdurre il termine nella faccetta corretta. Un maggiore aiuto da parte del sistema potrebbe venire evidenziando maggiormente, nella finestra di dialogo con l'utente, la faccetta di destinazione. Questo però non risolve il problema: l'utente pensa al suo b. i., non alla sua concettualizzazione e spesso non sa neppure che cos'è una faccetta.

Il secondo problema intrinseco nell'approccio booleano è che spesso una concettualizzazione del b. i. (individuazione delle faccette/concetti) si rivela non univoca durante la navigazione sul thesauro, in quanto più faccette possono essere fuse in una unica o, viceversa, una faccetta può essere scomposta in più faccette. Un esempio in cui si rende necessaria la fusione di due faccette è il seguente.

*Esempio 2*

Si consideri il b. i.: "Reperire *x* documenti che trattino il fenomeno della cooperazione nelle latterie". La query iniziale potrebbe essere costituita da 2 faccette, una contenente il termine 'latterie' e l'altra il termine 'cooperative'. In questo caso, il processo di riformulazione potrebbe procedere nel modo indicato qui di seguito:

```
query iniziale: (Latterie) & (Cooperative)
---(rif)--->
(Latterie,Latterie cooperative) &
(Cooperative,Latterie cooperative)
```

In questo caso, contrariamente all'Esempio 1, è corretto inserire il termine `Latterie cooperative` in entrambe le faccette (una latteria cooperativa è sia una cooperativa che una latteria). Qui le due faccette andrebbero fuse in:

```
(Latterie cooperative,(Latterie & Cooperative)).
```

Infatti, se si continua la riformulazione senza fondere le faccette si può ottenere:

```
---(rif)--->
(Latterie,Latterie cooperative,Cooperative) &
(Cooperative,Latterie cooperative,Latterie)
```

e questa query è certamente errata.

L'approccio attuale non sembra potere risolvere i problemi illustrati. Nella prossima sezione presento una possibile soluzione, basata su un approccio differente.

### 4.3. Soluzione dei problemi

Io penso che una causa del primo gruppo di problemi (termini proposti all'utente) sia l'approccio seguito per la riformulazione, ovvero scelta del focus e navigazione sul thesauro a partire da *un unico termine*. La strategia alternativa che propongo qui si rivelerà però utile anche per la soluzione del secondo gruppo di problemi (quelli intrinseci nei sistemi booleani).

Anziché scegliere il focus e navigare sul thesauro a partire da un unico termine, si può pensare di considerare *tutti* i termini della query (indico un generico termine della query con `Tq`) e *tutti* i termini nel thesauro in relazione[6] con almeno uno dei `Tq` (indico con `Tt` un generico termine del thesauro in relazione con un `Tq`). Inoltre, informazioni utili possono venire da altre 2 classi di termini: quelli estratti in qualche modo, ad esempio con operazioni come la `Zoom`, da documenti che l'utente ha giudicato rilevanti (termini che indico con `Td`) e quelli estratti allo stesso modo da documenti non rilevanti (`Tnd`); in questo modo si realizza quello che viene chiamato *relevance feedback*.

---

6 È necessario considerare anche i termini in relazione indiretta (raggiungibili percorrendo più di un arco sul thesauro). Per evitare problemi di efficienza temporale, bisognerà fissare una soglia per il numero massimo di archi percorribili.

Ora si può pensare ad una scelta dei termini da proporre all'utente (che indico con Tp) più fine di quella illustrata nella sezione 4.1, in quanto vi sono più criteri da adottare, rispetto alla riformulazione attuale, per effettuare tale scelta. Un Tp può infatti essere scelto fra i Tt e in Td, preferendo i Tp che non compaiono fra i Tnd e ordinando tali termini sulla base delle seguenti caratteristiche:

- attributi del Tp (posting count, flag controllato/non controllato, grado di interesse, ecc.);
- se il Tp appartiene al thesauro, attributi dei Tq con cui il Tp è in relazione;
- se il Tp appartiene al thesauro, numero di Tq in relazione con il Tp, numero di relazioni dei Tt con i Tq e tipo di tali relazioni;
- numero di volte che il Tp è stato proposto all'utente;
- eventuali pesi delle relazioni fra Tp e Tq in funzione della query iniziale (ad esempio, se il bisogno informativo è "*Influenza* del pascolo sulla qualità del latte", una relazione causes avrà un peso maggiore, almeno per quanto riguarda i termini derivati da pascolo);
- attributi delle faccette (posting count, grado di interesse, ecc.);
- se il Tp è un Td, frequenza di tale Td nei documenti rilevanti;

Uno schema riassuntivo, che schematizza anche l'algoritmo da seguire per implementare tutto ciò, è il seguente:

- costruire un insieme di termini contenente tutti i Tp; tale insieme è costituito da tutti i Tt e da tutti i Td;
- ordinare gli elementi di tale insieme sulla base delle caratteristiche indicate in precedenza, in modo che i primi termini dell'ordinamento siano quelli che hanno maggiore probabilità di essere confermati dall'utente. Questo può essere effettuato assegnando un *peso* ad ogni Tp, a seconda delle caratteristiche possedute;
- presentare all'utente solo i primi (secondo tale ordinamento) termini oppure presentare tutti i termini, però ordinati.

Per comprendere meglio la situazione, si consideri la rappresentazione grafica di figura 3: vi sono illustrate una semplice query booleana ((Latterie) & (Movimento cooperativo)), una piccola porzione del thesauro e l'indicizzazione di alcuni documenti.

La porzione del thesauro mostra come dovrebbe aver luogo la riformulazione. A partire dai termini che compaiono nella query, si percorrono gli archi del thesauro e ai termini man mano incontrati vengono assegnati dei punteggi a seconda delle loro caratteristiche. Ad esempio, il termine Latterie cooperative è in relazione con entrambi i termini della query iniziale e quindi gli vengono assegnati più punti rispetto, ad esempio, al termine Produzione lattea che è in relazione con un solo termine della query iniziale. Analogamente, lo stesso termine vede incrementato il suo punteggio in virtù del fatto che esso compare in un documento giudicato rilevante dall'utente, e così via. I termini con i punteggi più alti vengono infine proposti all'utente. Sulla base di considerazioni qualitative (ossia senza assegnare pesi ai termini),

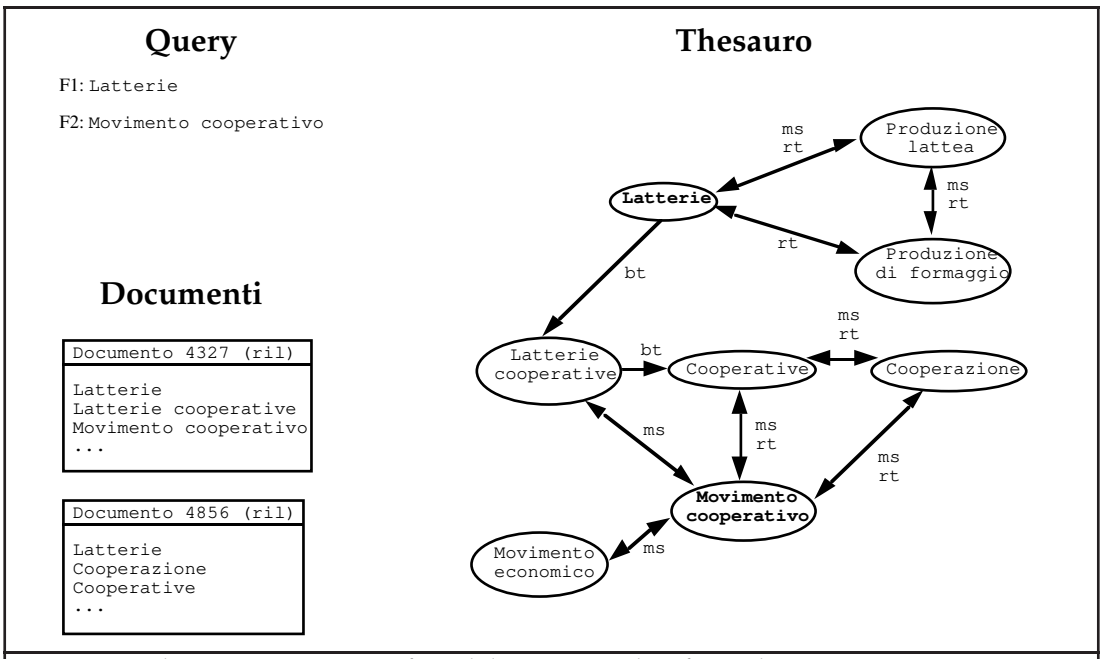


Figura 3. Schematizzazione grafica del processo di riformulazione proposto in questa sezione.

nell'esempio in esame verrebbero probabilmente proposti nell'ordine i seguenti termini: `Latterie cooperative`, `Cooperative`, `Cooperazione`, `Movimento economico`, `Produzione lattea` e `Produzione di formaggio`.

Giova ricordare tutte le informazioni utilizzate per preferire un termine ad un altro che in figura, per non complicare eccessivamente il disegno, non sono indicate: mancano gli attributi di termini e faccette, il numero di volte che i termini sono già stati proposti all'utente, frequenza dei termini nei documenti rilevanti, ecc.

I vantaggi rispetto alla riformulazione attuale sembrano evidenti: con l'approccio qui suggerito, si sfrutta una maggiore quantità di informazioni. Inoltre, man mano che la riformulazione procede, l'utente fornisce nuove informazioni, che vengono a loro volta sfruttate.

Vi sono però anche alcuni problemi, primo fra tutti la difficoltà a trovare esempi che dimostrino che l'approccio qui suggerito è consistentemente migliore di quello attualmente adottato in `FIRE`. A mio parere, tali difficoltà sono determinate dall'elevato grado di complessità del metodo proposto: vi sono numerosi fattori da tenere contemporaneamente in considerazione e una simulazione manuale è pertanto difficile. Alcune considerazioni a supporto di questa tesi:

- il numero di termini da considerare sale esponenzialmente in quanto bisogna fare più passi sul thesauro, a partire da vari termini e seguendo vari archi;
- se il numero di termini è alto, a maggior ragione lo è il numero di attributi di tali termini;
- il numero di `Td` e `Tnd` è elevato in quanto i documenti contengono parecchi termini;
- gli esempi più favorevoli si hanno in casi complicati, in cui sono presenti parecchie informazioni da sfruttare.

Un ulteriore problema è l'utilizzo di pesi e soglie, che sembra almeno di primo acchito inevitabile e che presenta alcuni punti discutibili, quali l'arbitrarietà dei valori da attribuire a pesi e soglie. Comunque, nella realizzazione di sistemi d'IA, tale strumento è ormai ampiamente usato e, oltre a ciò, un'implementazione alternativa non è da escludere a priori. Inoltre, le caratteristiche dei Tp vanno definite in modo più preciso e completo.

Un problema di carattere differente è quello che si presenta quando un Tp è in relazione con più Tq *in faccette diverse*. Come decidere in quale(i) faccetta(e) va aggiunto il nuovo termine? Questo problema si ricollega a quanto visto nella sezione 4.2: in certi casi è corretto aggiungere il nuovo termine in una sola faccetta, non sempre di immediata determinazione (Esempio 1 della sezione 4.2), mentre in altri casi il termine va aggiunto in più faccette (Esempio 2). Con l'approccio alla riformulazione proposto qui si possono utilizzare alcuni criteri per scegliere la faccetta in cui aggiungere un Tp accettato dall'utente:

- numero di relazioni del Tp con i termini di una faccetta;
- tipo di tali relazioni (ad esempio, una relazione nt come fra Produzione e Produzione di carne dovrebbe avere un peso superiore alle altre);
- assenza di relazioni con termini in altre faccette.

Anche qui, per implementare tutto ciò si può utilizzare un meccanismo a punteggi: il punteggio dell'assegnamento di un termine Tp ad una faccetta aumenta con il numero di relazioni fra Tp e i termini della faccetta; per ogni coppia termine-faccetta si può assegnare un punteggio e per ogni termine si sceglierà la faccetta con punteggio più alto. Se per un termine vi è una sola faccetta con punteggio significativo, non sussistono dubbi sulla scelta della faccetta in cui inserirlo; in caso contrario, può rendersi necessario fondere le due faccette (ovvero aggiungere il termine in entrambe).

Riconsiderando l'esempio illustrato in figura 3, si può osservare come il termine Latterie cooperative sia in relazione con termini delle due faccette: è un bt di Latterie ed è un ms di Movimento cooperativo. Per poter assegnare la faccetta corretta a tale termine è necessario che l'arco bt abbia un peso superiore rispetto a ms, e ciò risulta anche sensato da un punto di vista intuitivo: due termini, di cui uno più specifico dell'altro, vanno probabilmente usati entrambi per definire un dato concetto e vanno quindi inseriti stessa faccetta.

In questa sezione ho proposto un approccio alternativo alla riformulazione; esso sembra poter fornire risultati migliori rispetto alla strategia attualmente utilizzata e, almeno in prima analisi, si dimostra adeguato per la soluzione dei problemi evidenziati nella sezione 4.2.

## 5. Espressione del bisogno informativo in linguaggio naturale

Uno dei problemi riscontrati in FIRE (e in parte già incontrato nelle sezioni precedenti) è quello della mancanza di naturalezza intrinseca nella descrizione del b. i. dell'utente tramite il linguaggio del modello booleano. Infatti, un utente occasionale di un IRS, al contrario di un intermediario, spesso non sa che cosa sia una faccetta (o concetto) e fatica quindi ad

esprimere il proprio b. i. nel linguaggio interno dell'IRS. Questa limitazione è indubbiamente di grossa portata nel caso del sistema FIRE, in quanto una delle premesse fondamentali è quella di far interagire direttamente l'utente con il sistema, senza il tramite dell'intermediario. Sembra quindi irrinunciabile la possibilità per l'utente di definire il proprio b. i. in termini a lui più congeniali: il metodo più naturale è indubbiamente la definizione del b. i. in *linguaggio naturale*.

Se l'utente esprime il proprio b. i. in linguaggio naturale, vi è però il problema della traduzione nel linguaggio formale comprensibile dall'IRS.[7] Lo studio della comprensione del linguaggio naturale è ancora troppo arretrato per consentire la costruzione di un sistema sufficientemente generale; nel caso in esame qui è però forse possibile ottenere buoni risultati attraverso un procedimento che non richieda la comprensione del b. i.: come osserva Ingwersen (1992), non sempre un intermediario comprende la richiesta dell'utente.

È quindi possibile ipotizzare un processo di traduzione del b. i. espresso in linguaggio naturale in una query booleana in modo "non semantico":

(i) il b. i. espresso in linguaggio naturale è considerato come una sequenza di parole;
(ii) le parole appartenenti alla *stop word list* vengono eliminate (ad eccezione di quelle che compaiono all'interno di un'espressione comune[8] o un politermine presente nel thesauro), così come espressioni del tipo "Reperire *x*-*y* documenti che trattino";
(iii) si forma una prima query booleana in cui ogni faccetta è costituita da un unico termine o da un insieme di più termini se questi costituiscono un'espressione comune o un politermine presente nel thesauro;
(iv) vengono individuati termini appartenenti al thesauro;
(v) i termini rimanenti vengono associati tramite similitudine morfologica con termini appartenenti al thesauro, che vengono inseriti nella faccetta corrispondente;

A questo punto, il b. i. dell'utente è stato tradotto in una query booleana, sulla quale si può effettuare il consueto processo di riformulazione. In tale processo assumono un'importanza rilevante l'operazione di *fusione* di faccette, a cui ho già accennato nella sezione 4.2. Qui di seguito è illustrato come il processo presentato finora si concretizzi nel caso di un b. i. particolare:

b. i.: "Reperire 20-30 documenti che trattino il fenomeno della cooperazione nelle latterie da un punto di vista economico".

---

7 FIRE è basato su un sistema booleano, quindi la trattazione è qui limitata al caso del modello booleano. Comunque, parecchie considerazioni rimangono valide anche nel caso di un sistema basato sul modello *vector space*.

8 Il significato di 'espressione comune' è ovviamente da definire in modo più preciso. Un esempio, utilizzato più avanti, è l'espressione 'punto di vista', che contiene il termine 'di' il quale, nonostante appartenga alla *stop word list* non va eliminato.

(i),(ii),(iii)
```
(Cooperazione) &
(Latterie) &
(Punto di vista) & (Economico)
```

(Rif.)
```
(Cooperazione,Cooperative) &
(Latterie) &
(Punto di vista,Punto di vista economico) &
(Economico,Punto di vista economico)
```

(Fusione)
```
(Cooperazione,Cooperative) &
(Latterie) &
((Punto di vista economico),(Punto di vista & Economico))
```

(Rif.)
```
(Cooperazione,Cooperative,Latterie cooperative) &
(Latterie,Latterie cooperative) &
((Punto di vista economico),(Punto di vista & Economico))
```

(Fusione)
```
((Latterie cooperative),((Cooperazione,Cooperative) & (Latterie))) &
((Punto di vista economico),(Punto di vista & Economico))
```

Il b. i. espresso in linguaggio naturale viene tradotto in una query booleana, a partire dalla quale, con l'usuale processo di riformulazione (e utilizzando l'operazione di fusione di faccette) si ottiene la query finale che esprime in modo adeguato il b. i. iniziale. Bisogna spiegare che il termine `Punto di vista economico` compare nel thesauro, e ciò rende possibile la fusione effettuata come quarta operazione; lo stesso dicasi per il termine `Latterie cooperative` relativo all'ultima operazione.

Per raffinare ulteriormente questo procedimento, si può pensare di lavorare in parallelo su due query distinte, la prima ottenuta come illustrato poc'anzi, la seconda inserendo tutti i termini ricavati nel passo (ii) in un'unica faccetta: anziché inserire tutti i termini in *and* fra di loro, si considera l'*or* di tutti i termini. Si può poi procedere con il processo di riformulazione; qui sarà l'operazione di *scomposizione* di faccette a giocare un ruolo determinante.

Il problema dell'utilizzo del linguaggio naturale per l'espressione del b. i. dell'utente è indubbiamente importante per FIRE, ma potrebbe sembrare scollegato dagli altri problemi affrontati in questo documento; nella sezione 8 mostrerò invece l'esistenza di legami con gli altri argomenti.

## 6. Nuova interfaccia utente

L'interfaccia utente attuale di FIRE presenta alcune limitazioni, alcune già menzionate in precedenza, quali l'impossibilità della definizione del b. i. in linguaggio naturale o la difficoltà per l'utente inesperto a comprendere il processo di riformulazione (a causa della presenza delle faccette). In questa sezione, dopo aver richiamato il funzionamento dell'interfaccia attuale, propongo un'interfaccia utente basata su una filosofia differente e descrivo a grandi linee quale potrebbe essere, secondo me, l'aspetto dell'interfaccia utente nella prossima versione di FIRE; concludo poi con alcuni commenti sull'interfaccia proposta.[9]

[9] Le idee contenute nelle altre sezioni di questo documento sono quasi interamente mie. Al contrario, l'interfaccia utente proposta prende spunto da alcune idee del prof. Tasso e del dott. Brajnik, idee che ho solo parzialmente elaborato.

### 6.1. L'interfaccia utente di FIRE

Non illustro qui in modo completo l'interfaccia della versione attuale di FIRE, per la quale rimando a Floreanini (1993) e Brajnik et al. (1991b). La componente principale dell'interfaccia utente attuale è IRESFACE, che consente la costruzione della query (in formato booleano), la definizione di alcuni vincoli sulla ricerca, quali l'intervallo di documenti da reperire e l'obiettivo della ricerca (high-recall o high-precision), e la visualizzazione dei documenti reperiti.

L'altra componente fondamentale è il DSKBED, utilizzabile per visualizzare il thesauro. Nella versione attuale di FIRE questa componente gioca però un ruolo marginale (praticamente nullo) nell'interazione con l'utente: la riformulazione del b. i. avviene esclusivamente tramite IRESFACE, in cui vengono visualizzati i termini che IRES decide di suggerire all'utente.

Una critica da muovere alla versione attuale dell'interfaccia utente di FIRE è la sua rigidità: lo schema di interazione con l'utente è fissato a priori e le informazioni fornite all'utente non sono fornite nel modo più naturale. Si pensi al fatto che, quando il sistema suggerisce alcuni termini all'utente, l'informazione relativa al modo in cui tali termini sono stati ricavati (ovvero quali relazioni sussistono fra tali termini ed un termine della query) è espressa in forma *testuale*, mentre sarebbe sicuramente più comprensibile per l'utente la visualizzazione *grafica* di una porzione del thesauro. La stessa critica può essere mossa al DSKBED: neppure qui la visualizzazione del thesauro è effettuata in forma grafica.

Inoltre non vi è nessuno strumento che permetta di effettuare il *relevance feedback* dai documenti. È l'utente che deve esaminare i documenti alla ricerca di termini adeguati ad esprimere il suo b. i. ed inserire tali termini nella faccetta appropriata della query.

### 6.2. Proposta di una nuova interfaccia utente

L'interfaccia che propongo qui, basata su un approccio differente, cerca di colmare le lacune illustrate poc'anzi. L'aspetto della nuova interfaccia è illustrato in figura 4.

In tale figura si possono osservare le 4 finestre principali che compongono l'interfaccia: vi è una finestra per la definizione del b. i. in linguaggio naturale, una per la rappresentazione della query in forma booleana, una per la visualizzazione dei titoli dei documenti e una per visualizzare una *rete terminologica*. Le prime due finestre non necessitano di particolari spiegazioni e anche la finestra di visualizzazione dei titoli dei documenti è immediatamente comprensibile e simile a quella della versione attuale di FIRE (associata a tale finestra vi sono altre finestre, una per ogni documento, per visualizzare i singoli documenti su richiesta dell'utente). L'unica novità in tale finestra è la possibilità per l'utente di dividere i documenti in rilevanti e non rilevanti: ciò potrebbe essere effettuato ad esempio tramite due *cestini* (etichettati con "rilevanti" e "non rilevanti") in cui l'utente può trascinare i documenti dopo averli esaminati.

La finestra che necessita di maggiori spiegazioni è quella che visualizza la rete terminologica, ossia una rete in cui i nodi rappresentano termini e gli

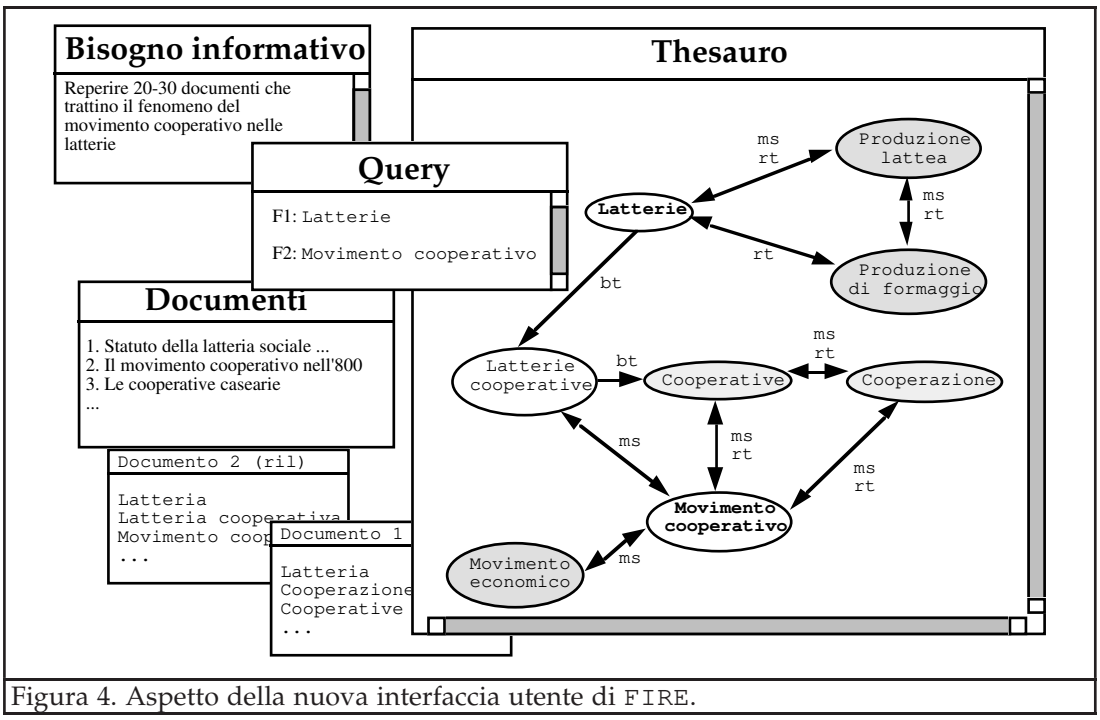


Figura 4. Aspetto della nuova interfaccia utente di FIRE.

archi relazioni fra termini. I termini possono appartenere al thesauro, ma questo non è necessario: essi possono essere ricavati ad esempio tramite *relevance feedback* dai documenti. Allo stesso modo gli archi possono rappresentare relazioni del thesauro, ma anche relazioni di tipo differente, quali la co-occorrenza di due termini nello stesso documento.

Nella rete terminologica compaiono i termini che costituiscono la query e i termini che il sistema suggerisce all'utente. Le varie categorie di termini sono individuate graficamente: i nodi relativi ai termini della query sono in grassetto, e lo sfondo dei termini che il sistema giudica più probabili hanno sfondo più chiaro rispetto a quelli meno probabili. Cliccando su un termine, l'utente può visualizzarne gli attributi (appartenenza o meno al thesauro, posting count, ecc.) o eliminarlo dalla rete terminologica (nel caso che tale termine non sia interessante per il suo b. i.).

La rete terminologica si modifica dinamicamente ad ogni operazione significativa dell'utente (modifica della query, giudizio di rilevanza su un documento, ecc.); per la sua gestione sarà quindi necessario un processo indipendente da quello che gestisce la finestra principale (quella della query). Tale processo dovrà agire in modo asincrono: quando la struttura dati (in comune fra i 2 processi) cambia stato, la finestra della rete terminologica viene modificata di conseguenza.

### 6.3. Alcuni commenti

Si osservi la similitudine della figura 4 con la figura 3 della sezione 4.3, relativa al nuovo approccio alla riformulazione: tale similitudine non è casuale, in quanto l'interfaccia utente proposta qui riflette numerosi aspetti del nuovo approccio alla riformulazione. Inoltre, è probabilmente improponibile la realizzazione di un'interfaccia di questo tipo utilizzando la versione attuale di FIRE, in quanto la rete terminologica va

necessariamente filtrata onde evitare di proporre all'utente un numero eccessivo di termini.

Se la rete terminologica è costituita da un numero limitato di nodi, le informazioni fornite all'utente sono chiare. Però, la situazione si complica se il numero di nodi aumenta. In tal caso, la rete può diventare difficile da gestire e comprendere per l'utente, e va quindi visualizzata in modo accorto. Inoltre, il processo che gestisce la rete è pesante computazionalmente (la rete da visualizzare cambia dinamicamente ad ogni operazione significativa dell'utente, quindi non si tratta semplicemente di visualizzare un grafo!); l'utilizzo di un processo dedicato alla gestione della rete dovrebbe comunque limitare questo problema.

Infine, alcune osservazioni sull'implementazione. Mentre la finestra per la definizione del b. i. in linguaggio naturale è una componente completamente nuova dell'interfaccia, le altre possono essere considerate rielaborazioni delle costituenti attuali: la finestra di visualizzazione della rete terminologica dovrebbe sostituire il DSKBED e le altre finestre sono, a parte alcune piccole modifiche, identiche a quelle presenti in IRESFACE. La difficoltà maggiore nell'implementare la nuova interfaccia è probabilmente costituita dal fatto che vi sono più processi che eseguono in parallelo, situazione che non è attualmente presente in FIRE.

## 7. Altre migliorie

In questa sezione discuto altre caratteristiche che sarebbero auspicabili in FIRE. Questa sezione, a mio parere necessaria per avere un quadro completo della situazione, sarà comunque molto meno approfondita delle sezioni precedenti: lo scopo è infatti di accennare ad alcune estensioni e ad alcune lacune che dovranno essere tenute in considerazione in sede di progetto della nuova versione del sistema.

Una prima limitazione a cui ho già accennato in precedenza è intrinseca nel sistema booleano: con tale modello è impossibile esprimere relazioni fra faccette. Un esempio di tale relazione potrebbe essere quello di relazioni di causa-effetto fra concetti, incontrato nella sezione 4.3 nel b. i. "Reperire *x-y* documenti che trattino l'argomento dell'*influenza* del pascolo sulla qualità del latte". Con il modello booleano, tale b. i. viene tradotto in una query costituita da 2 faccette in *and* fra di loro e rappresentanti rispettivamente i concetti 'Pascolo' e 'Qualità del latte'. La relazione di causa fra i due concetti viene quindi persa, mentre è una parte importante del b. i. dell'utente.

Un'altra estensione di FIRE potrebbe riguardare il meccanismo delle *spiegazioni*: il sistema non fornisce all'utente giustificazioni sul suo comportamento, o quantomeno lo fa in maniera limitata: ad esempio, non spiega in base a quali considerazione scelga un particolare focus, o perché talvolta, durante il processo di riformulazione, esegua una ricerca.

Vi sono parecchie informazioni che il sistema attuale non utilizza (anzi, neppure richiede all'utente) e che, secondo Ingwersen (1992), sono invece importanti. Fra di esse, si possono ricordare l'uso che l'utente intende fare con i documenti reperiti, il problema che l'utente intende risolvere con la ricerca bibliografica (*work space*) o il tipo di utente. Queste considerazioni inducono ad entrare nel campo dello *user modeling*: in FIRE un modulo di modellazione dell'utente (con tutta probabilità necessario in un IIRS, si

veda ad esempio Gri, 1993) è previsto, ma tale modulo non è integrato nel sistema.

Un'altra direzione in cui muoversi è l'implementazione delle operazioni di fusione e scomposizione di faccette: non sono affatto chiari i meccanismi su cui basarsi per decidere quando effettuare una di tali operazioni e uno studio a tale proposito potrebbe portare risultati interessanti.

Un altro campo di ricerca è quello della *specificità*, indice che indica per ogni termine del thesauro quanto specifico o generale sia (si veda Brajnik et al., 1991a). Essa si potrebbe dimostrare adeguata a risolvere un problema che si presenta comunemente, il reperimento di documenti troppo specifici o troppo generali rispetto al b. i. dell'utente. Collegato a ciò è forse un problema che si è evidenziato durante la sperimentazione della versione attuale di FIRE. Talvolta compaiono documenti molto generali, che trattano vari argomenti anche differenti fra loro (ad esempio atti di congressi o raccolte di articoli). Documenti di questo tipo andrebbero scomposti e, di volta in volta, solo le parti rilevanti per lo specifico b. i. andrebbero reperite.

Altre due limitazioni della versione attuale, ma che dovrebbero essere risolte a breve termine, sono la mancanza di un meccanismo di ranking dei documenti reperiti e di uno strumento analogo alla *Zoom* dell'ESA-Quest.

Infine, è in fase di realizzazione una valutazione del sistema FIRE, con l'utilizzo di una metodologia ricavata estendendo e migliorando quella descritta in Fornasier (1993) e Vidussi (1993). Il primo passo da effettuare in tale direzione è l'estensione della dimensione delle collezioni su cui FIRE lavora: a tale scopo si utilizzerà una collezione di 20.000 documenti ricavati dalla banca dati INSPEC.

## 8. Correlazioni fra gli sviluppi illustrati

Nelle sezioni precedenti ho esposto varie considerazioni sulle differenti direzioni di sviluppo del sistema FIRE. È vero che le varie direzioni di sviluppo per FIRE sono in buona parte indipendenti e, almeno da un punto di vista costruttivo, si possono affrontare indipendentemente. Però è anche vero che la realizzazione di un obiettivo può avere importanti influenze sugli altri. Alcuni legami sono già stati accennati man mano che se ne presentava la necessità; in questa sezione li richiamo e ne illustro di nuovi in modo da fornire un'esposizione completa e organica.

Il legame fra il thesauro stratificato (sezione 3) e il nuovo approccio alla riformulazione (sezione 4) è di triplice natura. Primo, queste due idee hanno entrambe l'obiettivo di migliorare la percentuale di termini accettati fra quelli proposti all'utente.

Secondo, con un thesauro stratificato il processo di riformulazione può fare affidamento su una maggiore quantità di informazioni (a causa della varietà di relazioni fra termini) e quindi fornire risultati migliori. Ad esempio, i pesi per i vari archi possono essere differenziati in modo più fine di quanto si può fare con un thesauro tradizionale: se si utilizza un thesauro stratificato, un arco di sinonimia avrà un punteggio più elevato di un arco `contrary`; in un thesauro standard, entrambi questi archi potrebbero essere codificati con una `rt` e quindi avrebbero lo stesso peso. Inoltre, lo sfruttamento di eventuali relazioni fra i concetti della query per assegnare pesi maggiori a certi archi del thesauro può essere effettuato (realizzando gli

opportuni metodi di navigazione sul thesauro arricchito) solo se il thesauro contiene archi non standard.

Terzo, un arco `isa` (presente in un thesauro stratificato) può essere utilizzato per decidere in quale faccetta inserire un termine confermato dall'utente in modo più corretto di un arco `bt` (l'unico utilizzabile in un thesauro standard), in quanto quest'ultimo potrebbe anche rappresentare una relazione `part-of`.

Anche la traduzione del b. i. in linguaggio naturale (sezione 5) può essere supportata dalla presenza di più tipi di relazioni nel thesauro (sezione 2). Riprendendo il solito esempio, se nel thesauro vi è una relazione di tipo `causes`, il termine (in linguaggio naturale) "influenza" nel b. i. può essere trattato nel modo corretto (ovvero, non facendolo diventare il termine di una faccetta, ma assegnando un peso più elevato agli archi `causes` del thesauro). Questo implica che il peso da assegnare agli archi del thesauro da percorrere durante la riformulazione non deve essere predeterminato: si può pensare ad un'assegnazione *dinamica* di tali pesi, in funzione del particolare b. i.

I legami fra le sezioni 4 (nuovo approccio alla riformulazione) e 5 (b. i. in linguaggio naturale) sono già stati accennati in precedenza. Il procedimento delineato nella sezione 5 implica infatti un ampio uso delle operazioni di fusione e scomposizione di faccette, e quindi richiede un particolare approccio alla riformulazione.

Come già accennato nella sezione 6, la rete terminologica, parte fondamentale della nuova interfaccia utente proposta in tale sezione, si ispira al processo di riformulazione della sezione 4. Inoltre, la rete richiede un adeguato filtraggio dei termini, onde evitare di sovraccaricare l'utente di informazioni e tale filtraggio è ottenibile con il nuovo approccio alla riformulazione.

Infine, il l'utilizzo di nuove relazioni fra faccette (sezione 7), la presenza di nuovi tipi di relazioni nel thesauro (sezione 3) e l'introduzione del b. i. in linguaggio naturale (sezione 5) sono evidentemente argomenti correlati.

## 9. Conclusioni

In questo documento, dopo aver brevemente presentato la versione attuale del sistema FIRE ho illustrato alcune idee che possono risultare interessanti nello sviluppo delle versioni successive di tale sistema. Ho proposto l'utilizzo di un nuovo tipo di thesauro (sezione 3), ho illustrato un nuovo approccio all'attività di riformulazione del bisogno informativo (sezione 4), ho evidenziato la necessità per l'utente di esprimere il proprio bisogno informativo in linguaggio naturale ed ho proposto un metodo di traduzione di tale bisogno informativo in query booleana (sezione 5), ho descritto l'aspetto di una nuova interfaccia utente (sezione 6), ho elencato alcune ulteriori lacune della versione attuale di FIRE e infine (sezione 8) ho evidenziato ed esposto organicamente i collegamenti fra le idee delle sezioni precedenti.

Alcuni dei miglioramenti proposti derivano dall'analisi dei problemi presentati dal sistema attuale, mentre altri sono basati su considerazioni più generali e intuitive. A mio parere è comunque importante tenere in

considerazione entrambi i gruppi in fase di progettazione della prossima versione del sistema.

Ho mosso numerose critiche al sistema FIRE attuale. Vi sono comunque alcuni aspetti (secondo me) positivi che è il caso di sottolineare, nell'ottica che questo documento serva da base per il progetto delle prossime versioni di FIRE: tali aspetti saranno infatti da mantenere anche in futuro.

Una prima qualità di FIRE è che è *supportivo*: il processo di riformulazione si limita a suggerire nuovi termini all'utente, il quale mantiene sempre il controllo della situazione; il sistema non modifica la query di sua iniziativa, al contrario di quanto avviene in altri IIRS. La supportività è considerata una caratteristica positiva da Ingwersen (1992).

L'approccio della riformulazione (tramite navigazione su thesauro e tecniche morfologiche) risolve un problema molto importante degli IRS: il problema del vocabolario (si veda Furnas et al., 1987). Inoltre, la riformulazione è un'attività realizzabile non solo per IRS basati sul modello booleano, ma anche, ad esempio, per IRS di tipo *vector space* o basati sulla logica *fuzzy*.[10]

## Bibliografia

Bates, M. J., 1979. Information Search Tactics, Journal of the American Society for Information Science, July 1979, pp. 205-214.

Brajnik G., Guida G., Mastrodonato L., Scaroni C., Tasso C., 1990a. *Progetto Generale del Sistema FIRE: Un Ambiente flessibile per lo Sviluppo di Interfacce Esperte a Sistemi di Basi di Dati Bibliografici*, Rapporto Tecnico CNR, n. 5/43, sottoprogetto Sistemi evoluti per Basi di Dati, Progetto Finalizzato CNR Sistemi Informatici e Calcolo Parallelo.

Brajnik G., Guida G., Mastrodonato L., Scaroni C., Tasso C., 1990b. *Interfacce Cooperative e Flessibili per Utenti di Basi di Dati Bibliografici*, Rapporto Tecnico CNR, n. 5/44, sottoprogetto Sistemi evoluti per Basi di Dati, Progetto Finalizzato CNR Sistemi Informatici e Calcolo Parallelo.

Brajnik G., Guida G., Mastrodonato L., Scaroni C., Tasso C., 1991a. *Specifiche di progetto del modulo per il retrieval intelligente IRES nell'ambito del sistema FIRE*, Rapporto Tecnico CNR, n. 5/61, sottoprogetto Sistemi evoluti per Basi di Dati, Progetto Finalizzato CNR Sistemi Informatici e Calcolo Parallelo.

Brajnik G., Guida G., Mastrodonato L., Scaroni C., Tasso C., 1991b. *Progetto dell'interfaccia utente del sistema FIRE*, Rapporto Tecnico CNR, n. 5/62, sottoprogetto Sistemi evoluti per Basi di Dati, Progetto Finalizzato CNR Sistemi Informatici e Calcolo Parallelo.

Croft, W. B., 1993. Knowledge-Based and Statistical Approaches to Text Retrieval, *IEEE Expert*, April 1993.

Danesi, D., 1990. *Le variabili del thesauro - Gestione e struttura*, IFNIA, Laboratorio Thesauri, Firenze.

10 Semmai, si può osservare che nel sistema booleano si presenta il problema aggiuntivo della scelta della faccetta in cui aggiungere un nuovo termine, problema assente, ad esempio, in un sistema *vector space*.

Floreanini, A., 1993. *Il Progetto FIRE - Studio e Realizzazione di un Sistema di Intelligent Information Retrieval*, Rapporto di Ricerca del Dipartimento di Matematica e Informatica dell'Università di Udine, Udine, Italy, rapporto n. UDMI/17/93/RR.

Fornasier, P., 1993. *Una metodologia di valutazione dei sistemi di Information Retrieval Intelligenti*, Tesi di laurea, Università di Udine, Udine, Italy.

Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T., 1987. The Vocabulary Problem in Human-System Communications, *Communications of the ACM*, v. 30, n. 11, November 1987.

Gri, F., 1993. *Analisi delle Funzionalità di Interfacce Intelligenti a Sistemi di Banche Dati Bibliografici*, Tesi di laurea, Università di Udine, Udine, Italy.

Ingwersen P., 1992. *Information Retrieval Interaction*, Taylor Graham, London.

Mizzaro, S., 1995. *La componente esperta del sistema FIRE*, Rapporto di Ricerca numero 1/95/RR del Dipartimento di Matematica e Informatica dell'Università di Udine, Udine, Italy.

Rich, E., 1986. *Intelligenza Artificiale*, Mc Graw-Hill, Milano, Italy.

Salton, G., 1989. *Automatic Text Processing – The Transformation, Analysis, and Retrieval of Information by Computer*, Addison-Wesley.

Vidussi, E., 1993. *Applicazione sperimentale di una metodologia valutativa al sistema FIRE*, Tesi di laurea, Università di Udine, Udine, Italy.