Università degli Studi di Modena e Reggio Emilia

Facoltà di Ingegneria di Modena
Laurea Specialistica in Ingegneria Informatica

Progettazione e sviluppo di un software per la visualizzazione di cluster di news

Relatore:
Prof. Sonia Bergamaschi

Tesi di Laurea di:
Diletta Leone

Correlatore:
Dott. Ing. Francesco Guerra

Anno Accademico 2007-2008

Indice

Elenco delle figure

Alla mia famiglia
e a mio nonno.

Fatti non foste
a viver come bruti
ma per seguir
virtute e canoscenza.

Introduzione

Il termine **Information Overload** si riferisce ad un fenomeno che accade usualmente agli utenti nel web: c'è troppa informazione a disposizione per prendere una decisione o per essere informati su un argomento. Il paradigma tradizionale, per il quale rimanere aggiornati significa raccogliere la maggiore informazione possibile, non sembra essere più valido. L'eccessiva informazione, infatti, distoglie l'attenzione dell'utente e induce il bisogno di allocare quell'attenzione efficientemente tra le molte fonti di informazione che la possono consumare. Internet fornisce più dati di quanto un umano sia capace di elaborarne e le applicazioni non sono in grado di supportare gli utenti nel ridurre l'information overload, fornendo strumenti efficaci per raccogliere, raggruppare, classificare, selezionare, indicizzare, ordinare e filtrare informazioni utili.

Una situazione analoga si verifica nell'ambito dei giornali online che pubblicano quotidianamente un grande quantitativo di articoli con contenuti parzialmente sovrapposti. Questo fatto genera information overload in una dimensione spaziale, quando articoli relativi allo stesso soggetto sono pubblicati da diversi giornali, e in una dimensione temporale, quando gli articoli relativi allo stesso soggetto sono pubblicati più volte in un breve periodo temporale. Dal punto di vista tecnico, e quindi dal punto di vista dei ricercatori in questo settore, l'obiettivo è quello di formalizzare ed implementare tecniche e metodi che siano sempre più performanti e permettano di ottenere risultati sempre migliori per quanto riguarda l'indicizzazione e il clustering di contenuti.

Questo documento riporta lo studio effettuato in relazione alle tecniche di clustering di

news, la classificazione e la valutazione delle tecniche e dei sistemi per il clustering di news sviluppati in letteratura, analisi delle tecniche specifiche per la selezione di news rilevanti per un utente, da integrare nel sistema RELEVANT^{News}.

Nel primo capitolo si analizzerà in dettaglio il fenomeno dell'information overload, focalizzando dapprima l'attenzione sui disagi che esso genera e successivamente analizzando le prime soluzioni al problema, ovvero motori sematici e aggregatori di news.

Nel capitolo secondo si parlerà del cluster analysis, si analizzeranno gli algoritmi di clustering di news usati in letteratura e i sistemi implementati valutando e confrontando le tecniche utilizzate e cercando di intuire i vantaggi che un utente potrebbe trarne.

Il terzo capitolo descrive dettagliatamente RELEVANT^{News}, sistema di clustering di news implementato dall'Università di Modena e Reggio nell'Emilia; vengono messi in rilievo gli algoritmi per il calcolo di similarità fra news e la possibilità che un utente ha di "giocare"sulle soglie di valutazione dei vari algoritmi presentati per ottenere metodi di raggruppamento differenti.

Il quarto ed ultimo capitolo presenta il lavoro svolto durante l'attività progettuale per estendere il software RELEVANT^{News}. Sebbene siano stati individuati diversi punti di lavoro (come ad esempio la possibilità di inserire un ulteriore criterio di scelta di clustering di news differente da quelli già implementati nel sistema), si è deciso di spostare l'attenzione sulla creazione di un'interfaccia grafica che riesca in qualche modo a rappresentare, in un layout radiale, la distribuzione dei cluster e delle proprie news nello spazio e le relazioni che intercorrono fra i diversi cluster.

Capitolo 1

Information Overload e News Overload

1.1 Cos' è l'Information Overload

Il **sovraccarico cognitivo**, meglio conosciuto come **Information overload(ing)**, si verifica quando si ricevono troppe informazioni per riuscire a prendere una decisione o sceglierne una specifica sulla quale focalizzare l'attenzione. La crescita esponenziale degli strumenti preposti all'invio e alla ricezione delle informazioni, la possibilità di quest'ultimi di effettuare connessioni alle reti locali, così come alle rete internet da una qualsiasi postazione, ha portato gli utenti ad una serie di possibilità infinite: alla capacità di reperire informazioni da ogni punto del pianeta, di essere in perenne contatto con i propri colleghi di lavoro, migliorando l'efficienza delle organizzazioni in maniera sostanziale. E' dunque facile comprendere come i modi con cui si può oggi venire in contatto con una persona siano molteplici, ma con l'aumento di tali sistemi di contatto, sono aumentati anche i modi per comunicare le informazioni superflue, indesiderate o anche messaggi pubblicitari, comunicazioni che quindi non sono richieste dall'utente, ma alle quali quest'ultimo è esposto.

Tutto ciò ha condotto quindi a quello che comunemente viene indicato con il termine *information pollution*. Ovviamente appena si fa riferimento al termine inquinamento informativo la nostra mente evoca la pratica dello spam, termine con il quale si rappresenta generalmente l'insieme di tutte quelle informazioni non richieste dagli utenti, ma di cui noi tutti siamo vittime involontarie.

Il sovraccarico di informazioni si riferisce, in sostanza, ad un eccesso di quantità di informazioni fornite, che spesso molti utenti fanno fatica a processare perché a volte non si è in grado di vedere dietro la validità delle informazioni stesse. Nel febbraio 2007 vi erano oltre 108 milioni di siti web che continuano tutt'oggi a crescere in modo esponenziale. Sempre più utenti hanno un ruolo attivo nel web, scrivono nei propri blog, rispondo nei forum, ed altri vengono considerati come spettatori, o come semplici partecipanti alla vita di Internet. Assistiamo a un sovraccarico di informazioni alle quali spesso si accede senza conoscere la validità dei contenuti e incorrendo nel rischio di disinformazione L'eccessiva informazione consuma attenzione. Quindi l'abbondanza di informazione genera una povertà di attenzione e induce il bisogno di allocare quell'attenzione efficientemente tra le molte fonti di informazione che la possono consumare.

Il fenomeno dell'information overload ha anche un impatto psicologico sull'internauta. Gli strumenti tecnologici che sono in grado di recapitare le informazioni agli individui, sono aumentati in maniera vertiginosa negli ultimi decenni, così com'è aumentata la possibilità di attingere al patrimonio informativo che tali strumenti generano; tutto ciò si contrappone però alle capacità umane di processare ed acquisire le informazioni, capacità che sono rimaste immutate nel tempo per chiari limiti di origine biologica del cervello umano. Nuove ricerche in ambito psicologico hanno dimostrato come gli individui incontrino svariate difficoltà nel compiere differenti operazioni nello stesso momento, basti pensare ad un tipico esempio, ovvero il dato inerente all'alto tasso di incidenti stradali dovuti all'uso del telefonino durante la guida; la possibilità degli individui di svolgere diversi compiti allo stesso momento, prevede un calo del livello di concentrazione che il

cervello dedica allo svolgimento di ogni singola attività. Questo si spiega semplicemente poiché il cervello umano non è in grado di trattenere diverse informazioni nel medesimo momento.

Studi empirici condotti da Torkel Klingberg, professore di *cognitive neuroscience* presso il Karolinska Institutet, hanno dimostrato come in un organismo umano, sottoposto a stress, siano presenti danneggiamenti nel sistema nervoso, immunitario, nel cuore e nel cervello [37]. Durante le sue ricerche Trokel ha sottoposto diverse persone a test sul livello di stress percepito da questi durante lo svolgimento della propria attività lavorativa. I risultati hanno dimostrato come i soggetti sottoposti a 20 e-mail indesiderate presentavano un grado di stress tanto elevato quanto i soggetti sottoposti alla visualizzazione di 100 e-mail indesiderate.

L'information overload purtroppo, non ha solo implicazioni psicologiche su un individuo, ma anche e sorpattuto delle implicazioni economiche.

Numerose ricerche sono state operate alla ricerca di una quantificazione del reale costo che l'information overload procura all'economia globale. Negli ultimi anni si sono affermati diversi gruppi di ricerca che hanno segnalato come i costi imputabili all'information overload siano in rapida ascesa, e come tali costi abbiamo effetti sia diretti, che indiretti sull'attività economica delle imprese.

Tra la fine del 2007 e l'inizio del 2008 è stata fondata la Information Overload Research Group (IORG), un'organizzazione no-profit che si occupa del problema dell'IO; tale associazione ha compiuto alcuni passi in avanti nell'osservazione delle dinamiche che le aziende stanno innescando per combattere la rapida ascesa di tale fenomeno. Anche diverse testate giornalistiche si sono occupate dell'Information Overload, in primis il New York Times che ha pubblicato, qualche tempo fa, una ricerca condotta da Basex (la principala azienda in ambito di ricerca sull'economia della conoscenza al mondo), che si riferiva ad una stima dei costi economici di tale fenomeno, cifra vertiginosa che era stimata intorno ai 650 miliardi di dollari in riferimento alle previsione per l'anno

2008, in realtà le stime attuali prospettano una situazione ancora peggiore: si è calcolato infatti che nel 2009 l'IO porterà all'economia mondiale costi che si aggirano inorno ai 900 miliardi di dollari, cifre che riflettono costi in termini di perdita di produttività e riduzione dell'innovazione.

I risultati di numerose ricerche hanno dimostrato che molti utenti impiegano il 28% del loro tempo lavorativo, nel cercare di individuare le informazioni necessarie allo svolgimento delle proprie attività lavorative, ciò comportava quindi 28 Miliardi di dollari persi in termini di costi sostenuti per il pagamento del lavoratore durante l'anno, ciò moltiplicato per il salario medio percepito, ha condotto all'ottenimento delle cifre sopracitate.

E' da poco stato reso disponibile sempre da Basex uno strumento applicativo sul Web che indicare a quanto ammonta il costo dell'Information Oveload, per scoprirlo basterà indicare il settore di riferimento, il numero di lavoratori e la tipologia degli stessi (da highly skill ad unskilled). Ma le stime e i possibili calcoli a cui sarebbe possibile giungere, non prendono in considerazione alcuni danni collaterali che sono causati anche alla salute del management e degli altri lavoratori che sono sottoposti ad un tale sovraccarico, inoltre questo tipo di fenomeno causa loro oltre ad un dispendio inutile di energie, anche un forte stress emotivo, fattori che si ripercuotono sulla condizione psico-fisica dei lavoratori che si trovano a dover fare i conti con questa problematica.

Tutte le ricerche effettuate convergono in un solo punto: l'utente è incapace di individuare,nelle proprie circerche, una variabile che possa scremare le diverse informazioni che giungono tramite i diversi canali. Per questo motivo le informazioni giungono in modo indiscriminato e con sistemi di filtraggio spesso inefficaci [15]. Gli effetti economici non sono i soli effetti che sono generati dall'IO.

Come è possibile individuare dai dati in tabella **??** la maggior parte delle risposte indica una perdita di tempo che si riflette direttamente su una riduzione della produttività, dell'efficienza e di conseguenza rispecchia gli effetti negativi che si ripercuotono

Effect	Number	Percentage
Loss of time	87	72%
Negative effect on work	48	40%
Reduced efficency	19	16%
Frustration, tiredness, stress	19	16%
Negative effect on decision quality	16	13%
Reduced productivity	10	8%
Effect on department or whole organization	9	7%
Damage to personal life	4	3%
N0ne	4	3%

Tabella 1.1: Effects of information overload

sull'intera organizzazione. Oltre a ciò possiamo vedere come gli effetti si ripercuotano anche attraverso patologie quali la frustrazione, un senso di stanchezza e lo stress emotivo che gli intervistati subiscono dalla sovraesposizione a messaggi ridondanti o non critici per il loro lavoro, le patologie sovraelencate influiscono negativamente anche sulla loro capacità di effettuare decisioni, ciò compromette in tal modo l'ambiente lavorativo in cui essi si trovano ad operare.

Come è facile intuire, una fra le maggiori fonti di informazione è Internet che mette a disposizione più nozioni di quante un utente in realtà ne riesca a processare e i software attualmente in uso, non riescono a far fronte alla riduzione del sovraccarico di informazioni e sotto tale ottica, bisognerebbe collezionare, raggruppare, selezionare o indicizzare e filtrare le informazioni realmente utili [20]. Dunque si può affermare che a tutt'oggi, non esiste un modo, che sia universalmente riconosciuto, per ottenere una qualità informativa perfetta.

In base al sondaggio che abbiamo citato in precedenza, possiamo identificare diversi sistemi di filtraggio, quello più diffuso è l'uso dei filtri informatici, in alternativa sarebbe

Category	Solutions	Number	Percentage
P1	Filter the information	59	47%
P2	Eliminate the source	30	24%
P3	Delegate work	30	24%
P4	Prioritize	22	18%
T1	Utilize technology	17	14%
O1	Organize work	14	11%
O2	Enhance communication	10	8%
O3	Other	10	8%
O4	Consult top management	9	7%
O5	Help from IS/IT department	8	6%
I1	Ignore information	5	4%

Key: P = personal T = technological O = organizational I = ignore

Tabella 1.2: Solutions to information overload

possibile procedere alla eliminazione delle fonti che generano i messaggi sotto accusa, un'altra soluzione che veniva proposta nella ricerca era i sistemi di delega al personale di segreteria, che si preoccupa di individuare i messaggi critici ed inoltrarli poi ai manager, la tabella **??**, indica in tal senso i sistemi che sono maggiormente adottati per la risoluzione del problema.

Riuscire a controllare il fenomeno dell' information overload ha portato anche alla sperimentazione di motori di ricerca semantici. A differenza dei tradizionali motori di ricerca, definiti sintattici, che si preoccupano di "censire" le parole che ci sono all'interno di un testo - le keyword - senza in alcun modo tentare di determinare il contesto in cui queste parole vengono utilizzate, la ricerca semantica tenta invece di avvicinarsi al meccanismo di apprendimento umano: il lettore non memorizza le singole parole, bensì tenta di sviluppare una " mappa cognitiva" che gli consenta di estrarre il significato di

quanto sta leggendo. Occorre quindi analizzare il testo in maniera molto simile a quanto fanno le persone, interpretando il significato logico delle frasi e tentando di carpirne il significato dal contesto. Un procedimento di apprendimento complesso, soprattutto per un computer, e solo di recente la tecnologia si è evoluta al punto da renderlo possibile: ma si tratta di una attività che richiede investimenti e capacità, in cui è difficile improvvisare.

In realtà, quelli della ricerca semantica, sono "obiettivi ambiziosi" occorreranno forse altri anni per consentire lo sviluppo di una tecnologia di ricerca semantica in grado di divenire pervasiva e orizzontale. Allo stato attuale consente di offrire strumenti per compiti precisi: categorizzare la posta elettronica, effettuare ricerche ristrette a settori specifici della conoscenza.

1.2 Semantic search engine

Implementare un motore di ricerca semantico, peraltro, non è una procedura semplice, richiede molto lavoro per adattarlo ad ogni lingua e cultura. Quando si lavora con un approccio simbolico non cambia molto se la parola usata per dire tavolo è l'inglese table, il tedesco tisch. Se subentra la tecnologia semantica, il computer necessita di essere istruito poichè l'aspetto linguistico conta: gli stessi oggetti si possono indicare con termini differenti e ci sono concetti che non esistono neppure in culture differenti, o che vengono indicati con parole diverse.

I passaggi fondamentali per costruire un motore semantico non sono molti [16]. Prima di tutto occorre capire che tipo di informazione deve essere trattato, vale a dire quale tipo di conoscenza debba essere gestita: manuali di auto, programmi tv etc. Occorre inoltre specificare la granularità delle news, ovvero si vuole una informazione precisa e puntuale o informazioni generiche. Sono passaggi che hanno un ruolo di notevole importanza nell'interazione con l'utente.

A questo punto si passa alla parte operativa: il software va alimentato con la conoscen-

za del sistema, in modo da permettergli di individuare concetti ed espressioni problematiche: in un manuale automobilistico alcuni concetti come volante, sedile, cambio, saranno conosciuti e riconoscibili: ma in un sito che parla di persone e show televisivi, i titoli di questi ultimi potrebbero risultare un problema.

I motori di ricerca sono quindi uno strumento prezioso nelle mani dell'utente che si serve di Internet per cercare dati, informazioni e documenti. La nuova frontiera, rappresentata dai motori "intelligenti", oltre a selezionare le pagine web, ordinandole per rilevanza, le classificano per argomento, suggerendo i percorsi di ricerca e di approfondimentoin base all'argomento desiderato e supportando il ricercatore come farebbe un esperto del settore. A fronte di evidenti vantaggi, i motori di nuova generazione presentano infatti anche alcuni rischi che, se tenuti in debito conto, possono essere fronteggiati e superati, in modo da poter trarre il massimo beneficio da strumenti di ricerca innovativi e utili in vari campi, dall'economia alla finanza, dal marketing all'informazione e alla documentazione.

I motori di ricerca rappresentano lo strumento principale per l'utente che desidera cercare dati o informazioni in Internet. Dai dati di mercato alle analisi di settore, dalle informazioni varie su fatti, problemi e persone fino agli argomenti più futili e giocosi: in Internet c'è di tutto. Il problema è come trovare le informazioni necessarie per soddisfare una determinata domanda, cercando di districarsi in un *mare magnum* d idati spesso fuorvianti, ridondanti o comunque troppo numerosi per essere gestiti con facilità.

I motori di ricerca svolgono due attività principali. Prima di tutto navigano nella rete per trovare informazioni. Per fare questo si avvalgono di applicazioni automatiche, detti spiders (ragni) oppure anche crawlers (cioè nuotatori). Questi non sono altro che programmi che partono da un insieme di URL e seguono la struttura ipertestuale del web per accedere ai documenti disponibili, generando poi un indice dei termini in essi contenuti. Attraverso delle tecniche di indicizzazione, associano a ogni URL un coefficiente di rilevanza, ossia una specie di misuratore di quanto un particolare URL può

essere importante rispetto a un dato termine. Il secondo compito dei motori di ricerca consiste nel rispondere alle richieste da parte degli utilizzatori. Infatti propongono in ordine di importanza le risorse presenti online che possono essere utili all'utente che ha introdotto nel motore un dato quesito.

In definitiva possiamo identificare due tipi di motori di ricerca: il primo è un semplice indice di argomentiche legge esclusivamente i titoli e le descrizioni; il secondo utilizza il sopracitato spider, cioè un programma di indicizzazione in grado di restituire risultati molto più selettivi.I motori hanno contribuito a dareun ordine alle innumerevoli risorsedella rete, creando vasti archividi dati che comprendono un grannumero di pagine web. Alla complessitàdel metodo di archiviazionedei dati adottato dai motori,corrisponde la facilità d'uso da parte dell'utente.

Ai motori di ricerca si affiancano le directory, che consistono in grandi archivi di siti, selezionati da personale specializzato e proposti al ricercatore in un indice di categorie. Le directory più note in Italia sono, ad esempio, Virgilio (www.virgilio.it), Yahoo! (http://it.yahoo.com/) e DMOZ (http://www.dmoz.org/World/Italiano/).

Ogni categoria si suddivide in sottocategorie, che a loro volta hanno altre sottocategorie. L'utente accede alla categoria d'argomenti di interesse e può affinare la ricerca selezionando le varie sottocategorie che gli vengono proposte. Le risorse on line riportate nelle varie directory vengono scelte dagli operatori umani che danno vita alle directory stesse, e quindi non derivano da una scansione automatica e continua di tutto il contenuto della rete (come, invece, avviene per i motori di ricerca che si avvalgono, per questa attività, di robot).

Le directory sono utili soprattutto quando l'utente ha ben chiaro checosa vuole chiedere e che cosa vuole ottenere dalla rete. La sfida per il futuro di Internet è di dotare i motori di ricerca di un'intelligenza che ancora non hanno. La strada aperta da alcuni nuovi motori porta verso un perfezionamento della ricerca del documento giusto che soddisfi il più possibile le esigenze dell'utente. Per avere questo occorre una lettura

dei documenti presenti in rete simile a quella che farebbe un essere umano: i documenti vengono analizzati non solo nella forma, ma soprattutto nel loro contenuto, tramite regole, inferenze e definizioni, utilizzando criteri semantici e concettuali. Il punto consiste proprio nel definire categorizzazioni, classificazioni, relazioni, schemi, associazioni, collegamenti fra dati e informazioni.

In questo modo il motore di ricerca del futuro (che, come vedremo, sta diventando già una realtà del presente) si allontana dalla ricerca per indirizzi e parole chiave, per andare nella direzione di una ricerca semantica basata su concetti e categorie. Si tratta di un valore aggiunto offerto all'utente, che in questo modo viene supportato nel reperimento delle informazioni giuste adatte al quesito sottoposto al motore di ricerca.Questi nuovi motori si basano sul document clustering, ossia sulla classificazione dei documenti, che vengono scandagliati nei contenuti e proposti suddivisi per argomento (appunto per classificazione) e per rilevanza.

In questo modo l'utente ha un aiuto in più per capire se e come le pagine web trovate dal motore sono per lui più o meno interessanti. L'utente di Internet può incontrare difficoltà nel reperire in tempi ragionevoli ciò che gli serve, perchè spesso non è in grado di sfruttare al meglio gli strumenti che il web gli mette a disposizione. Se usati in modo corretto, i motori di ricerca guidano l'utente fino al risultato atteso. Per ricerche più complesse, o per quesiti non ben definiti, sono necessari strumenti aggiuntivi: classificare i dati e i documenti presenti in rete come farebbe una persona esperta di tutto il contenuto del web è una caratteristica che può fare di un motore di ricerca uno strumento prezioso in mano al ricercatore più esigente (e magari anche per quello meno esperto nella navigazione in rete).

Tutto questo si inserisce nel più ampio discorso dell'intelligenza artificialee del cosiddetto "machine learning", cioè l'apprendimento automatico. Da ciò derivano il data mining, il text mining e il web mining, che trovano applicazione proprio nel settore della classificazione dei documenti presenti in Internet. Chiarire questi concetti, aiuta anche

a comprendere il quadro di riferimento dello stesso document clustering.

Il data mining, che consiste nel processo di estrazione di conoscenza da banche dati di grandi dimensioni attraverso l'applicazione di algoritmi che individuano le associazioni "invisibili", o comunque nascoste, tra le informazioni e le rendono quindi visibili. In questo modo vengono esplorate grandi quantità di dati e le informazioni di maggiore rilievo e interesse vengono identificate, isolate e rese disponibili.

Questo procedimento è anche definito "estrazione di conoscenza" e avviene attraverso il reperimento di associazioni e di sequenze ripetute nei dati. Così queste associazioni indicano una struttura o, più in generale, una rappresentazione sintetica dei dati. Questa procedura non è scevra da rischi impliciti, come ad esempio trovare correlazioni che nella realtà o non esistono o non sono effettivamente significative. In definitiva il processo di datamining offre risultati apprezzabili solo in seguito a una attenta interpretazione dei risultati ottenuti.

Se integriamo il data mining nell'ambito della linguistica, parliamo allora di text mining. Questo procedimento consiste nell'estrazionee nella mappatura di informazioni direttamente dai testi. In questo modo si può realizzare una sorta di mappa cartografica delle informazioni. Tale attività può essere messa a buon frutto nelle ricerche in Internet,e in particolare nei documenti presenti nel web: infatti si tratta di una specie di "filtraggio intelligente" di documenti in base alle esigenze specificate dall'utente.Si stima che la maggior parte delleinformazioni presenti in rete è rappresentata da testi: da ciò si comprende l'importanza strategica che il text mining può assumere, soprattutto in ambito economico-commerciale.

Se applichiamo insieme il data mining e il text mining abbiamo ilcosiddetto web mining, che consistenella ricerca di associazioni sulpiano dei contenuti, della struttura e dell'uso delle informazioni. I contenuti vengono studiati prendendo in considerazione i dati raccolti dai motori di ricerca e dai web crawlers. La struttura viene esaminata partendo dai dati che riguardano la struttura stessa di una specifica pagina web. L'uso

viene analizzato in base ai dati relativi a un determinato browser. Una volta ottenute le informazioni con il web mining, si procede a un' ulteriore valutazione, spesso attraverso l'utilizzo di alcuni parametri del data mining, come il cosiddetto clustering, quindi ricercando e definendo le possibili aggregazioni, classificazioni e associazioni tra i dati.

Le possibili (ed effettive) applicazionisono numerose, soprattutto nei settori del marketing, delle indagini di mercato e della gestione aziendale. Nell'ambito dell'informazione si applica il processo di text mining. Alla base di alcuni motori di ricerca troviamo gli stessi algoritmi utilizzati per il text mining: permettono di ricercare i dati e proporli all'utente suddivisi per categorie.

Come si è già accennato, per document clustering si intende il processo di raggruppamento delle pagine e dei documenti trovati nel web secondo pattern semantici, parole chiave e temi.

Si tratta di una modalità di presentazione dei risultati di ricerca, utilizzata dai motori di ricerca di nuova generazione. Secondo questa modalità, il motore non offre solo, come risultato della ricerca, l'elenco delle pagine web più significative in base alla domanda inserita dall'utente, ma presenta anche un elenco di pagine web classificate per argomenti attinenti all'oggetto della ricerca dell'utente.

Quest'ultimo viene così consigliato su come indirizzare la propria ricerca grazie a una prima classificazione delle pagine web effettuata dal motore di ricerca stesso. Riportiamo alcuni esempi di motori semantici.

Vivìsimo - http://vivisimo.com -

Fondato nel 2000 da alcuni ricercatori della Carnegie Mellon University, può essere definito un "motore per il raggruppamento di documenti" [16].

Utilizza un algoritmo di clustering per organizzare i risultati della ricerca in categorie e visualizzarli anche per gruppi tematici, oltre che in ordine di importanza e di argomento. Vivìsimo è utile soprattutto quando l'utente ha bisogno di farsi un'idea di un argomento o semplicemente di un termine di cui non conosce nulla. Inoltre è utile per trovare tutti

i possibili termini e i concetti correlati all'argomento. La definizione corretta sarebbe clustering engine, cioè uno strumento che raggruppa le risorse della rete su un dato argomento, rendendole fruibili attraverso cartelle tematiche create in tempo reale. Al momento Vivìsimo è disponibile solo in lingua inglese.

Teoma -www.teoma.com-

In gaelico Teoma significa "esperto". Nato come progetto sperimentale nel 1998, è stato acquisito nel 2001 dalla Ask Jeeves Inc., la società che gestisce il più noto motore di ricerca Ask Jeeves.

Questo motore di ricerca raggruppa in tre sezioni i risultati della ricerca: un gruppo tematico di cartelle (Web pages grouped by topic); i singoli indirizzi web correlati di una breve descrizione, (Web pages); una serie di link a siti specializzati sull'argomento richiesto (expert's links).

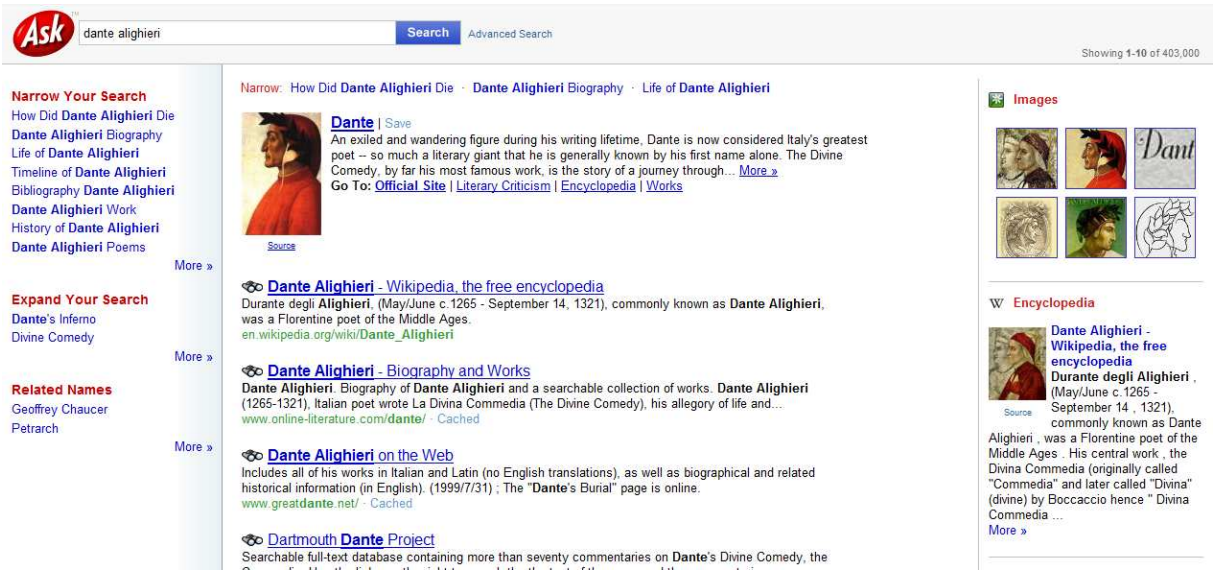

Figura 1.1: Teoma Semantic search engine.

Teoma è molto simile al più famoso Google, in particolare per quanto riguarda l'algoritmo di ricerca, e punta a definire i risultati di ricerca in base alla validità delle pagine web riguardanti lo stesso argomento. A differenza di Google, come accennato sopra, viene

offerta la possibilità di affinare ulteriormente il risultato della ricerca. Teoma punta alla qualità, sia dei siti indicizzati sia dei risultati di ricerca. La stessa novità di proporre link di esperti (expert's links), messi in rete da persone o gruppi di appassionati o esperti in un determinato ambito, ne è una prova. Teoma offre anche la possibilità di effettuare una ricerca avanzata, compilando un form con varie specifiche per circoscrivere il più possibile l'ambito della ricerca. Anche Teoma è disponibile solo in lingua inglese.

WiseNut - www.wisenut.com -

È lanciato sul mercato insieme a Teoma, nel 2001. Ha un'interfaccia molto semplice, sulla falsariga di Google. Creato in realtà nel 1999, utilizza un sistema di ordinamento delle pagine web in base alla rilevanza. Questo algoritmo misura l'importanza sia delle pagine web in generale sia di quelle trovate all'interno della ricerca. Inoltre valuta il contenuto delle pagine web, analizzando i link riportati e la loro provenienza. La tecnologia adottata è molto simile a quella impiegata da Teoma. Offre la possibilità di effettuare la ricerca in 25 lingue, compreso l'italiano. Il risultato della ricerca effettuata con WiseNut è duplice.

Da un lato il motore propone la lista delle pagine web scelte: per ognuna viene indicato un breve riassunto del contenuto e, grazie alla funzione "Sneak-to-peek", viene consentito di "sbirciare" la pagina web proposta senza bisogno di aprirla completamente ma semplicemente visionandone un'anteprima. Inoltre WiseNnut mostra un breve elenco di categorie che riuniscono altri link ad altre pagine relative all'argomento della ricerca. In alcuni casi offre la possibilità di approfondire la ricerca proprio attraverso una di queste categorie. Se si utilizza WiseNut, la possibilità di effettuare le ricerche in lingua italiana su pagine web italiane è un valore aggiunto che manca agli altri motori di nuova generazione, anche se la funzione di clustering risulta meno accurata.

Clusty - www.clusty.com -

È un metamotore sviluppato da Vivìsimo e si basa sulla funzione di clustering dei risultati di ricerca. Infatti il suo nome deriva proprio dal termine cluster. È stato lanciato

sul mercato con la versione beta nel settembre del 2004, dopo uno sviluppo durato quattro anni. Clusty aggiunge alcune nuove caratteristiche e una nuova interfaccia rispetto a Vivìsimo.

Ad esempio offre la possibilità di effettuare ricerche anche tra i blog e tra le news.

Inoltre è interessante la possibilità di personalizzare, da parte dell'utente, le modalità di ricerca. Ogni pagina web trovata può essere aperta in anteprima, senza dovere accedere al link per visionarne il contenuto. Inoltre è attiva una funzione che rimanda una determinata pagina web trovata nelle classificazioni proposte per la stessa ricerca. Infatti anche Clusty propone una serie di siti classificati per argomenti affini a quello della ricerca avviata dall'utente. Queste classificazioni possono essere ordinate (e mostrate direttamente) in base all'argomento, alla fonte da cui sono state tratte le pagine web scelte, agli URL. Si può scegliere di effettuare la ricerca su ad esempio Dante Alighieri in varie sezioni di Clusty: "Web", "News", "Images", "Shopping", "Encyclopedia", "Gossip". Basta inserire una volta sola nella stringa di interrogazione le parole "dante alighieri" e poi scegliere di volta in volta la sezione che ci interessa.

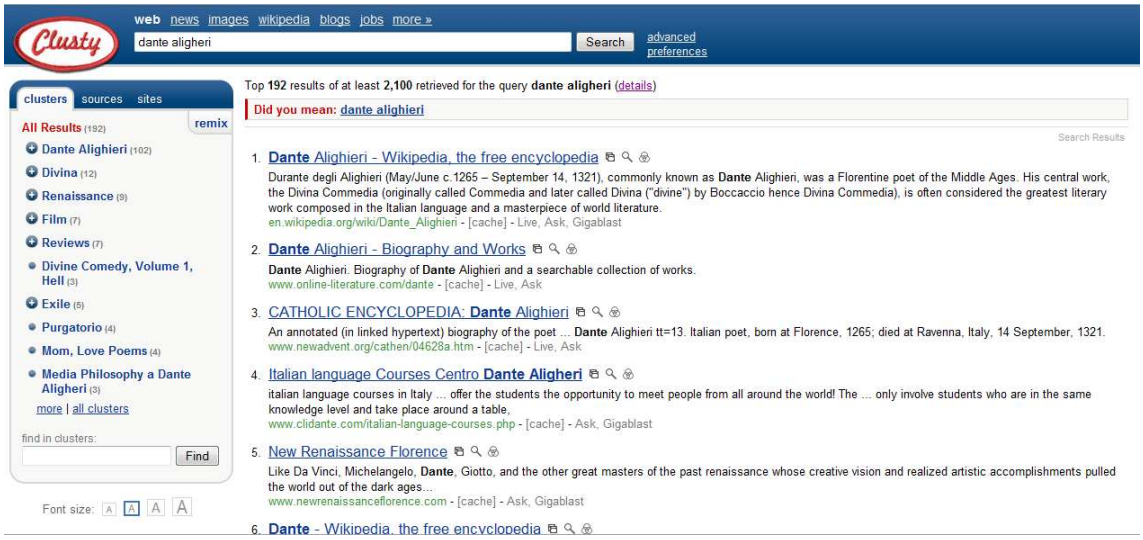

Figura 1.2: Esempio di semantic search engine.

Anche Clusty è disponibile solo in lingua inglese, ma offre ampie possibilità di ricerca

e di approfondimento. Si presenta come uno strumento agile, veloce e duttile.

Turbo10 - http://turbo10.com-

È un metamotore sviluppato da Fleetfoot Internet Solutions Limited (UK). Sfrutta un algoritmo di clustering: in base all'argomento, interroga la directory o il motore di ricerca che identifica come essere più adatto, e mostra in fondo alle classificazioni (clusters) la risorsa utilizzata per trovarli.

Quindi, oltre ai motori di ricerca, interroga anche numerose directory: il vantaggio consiste in una maggiore precisione della ricerca e nel reperimento di un maggior numero di documenti. Inoltre Turbo10 offre la possibilità di creare una collezione personalizzata di directory o anche di motori di ricerca adatti alla esigenze dell'utente. In più suggerisce un motore da aggiungere alla lista nel caso in cui non fosse già presente. La caratteristica interessante di Turbo10 è che permette all'utente di ordinare manualmente le voci di classificazione e i documenti in base al livello di pertinenza e di rilevanza. È presente anche la funzione "Search-O-Meter", che consente di muoversi da una pagina all'altra, da un cluster all'altro, mettendo in evidenza i documenti già visionati.

Anche nel caso della nostra ricerca "dante alighieri", Turbo10 visualizza nella parte centrale della pagina dei risultati le pagine web scelte, ma più interessante risulta essere la classificazione per argomento e l'indicazione dei motori utilizzati per ottenere i risultati più pertinenti.

Lo svantaggio di Turbo10 consiste nel fatto che, ancora una volta, è disponibile solo la versione in inglese e la ricerca viene effettuata di preferenza su pagine web di lingua inglese.

Kart00 - www.kartoo.com -

È un metamotore di ricerca che ha la particolarità di proporre i risultati sotto forma di mappe grafiche bidimensionali o tridimensionali. I siti web trovati vengono infatti rappresentati con icone più o meno grandi in base al grado di rilevanza.

Per affinare la ricerca, l'utente viene guidato dalla mappa stessa all'utilizzo di parole

chiave. I risultati della ricerca possono essere filtrati. Le mappe possono contenere dei siti cosiddetti "parasiti", cioè non concordanti con l'oggetto della ricerca.

Per escluderli, esiste un apposito pulsante che serve per eliminarli dai propri risultati.

Kart00 offre la possibilità di scegliere la lingua. I risultati della ricerca vengono disposti in una mappa: le icone che appaiono, quando si passa sopra con il cursore del mouse, mostrano le parole chiave corrispondenti e, a sinistra della pagina, appare una breve descrizione del sito. A questo punto è possibile affinare la ricerca, aggiungendo o escludendo dei temi.

Interessante è l'indicatore a forma di barometro che rappresenta graficamente il numero di siti che corrispondono alla ricerca. I cluster si presentano come parole bianche su sfondo blu, disseminate sulla superficie della mappa e collegate tra di loro. La vera novità di Kart00 consiste nell'interattività della mappa: infatti spostandosi con il cursore del mouse sopra le varie zone della mappa, possiamo creare vari livelli di legami tra le informazioni che il metamotore ha selezionato nel web per noi. In un esempio di ricerca, i legami tra le risorse trovate sono numerosi e tutti rilevanti. L' interfaccia visuale, davvero innovativa, rappresenta veramente una marcia in più per Kart00, che, inoltre, possiede tutte le caratteristiche dei migliori motori di nuova generazione.

I motori di ricerca si muovono da sempre nella rete mondiale, effettuando le loro ricerche per indirizzi e per parole chiave.

La nuova generazione invece cambia prospettiva, e va verso un' analisi per concetti, per categorie, addentrandosi nella ricerca semantica. Grazie a questi nuovi strumenti di ricerca, l'utente viene guidato verso un ampliamento della propria conoscenza, accompagnato attraverso proposte di navigazione, aiutato da interfacce grafiche facili, immediate e addirittura interattive. Il lavoro dei motori di ricerca diventa sempre meno meccanico e sempre più simile al contributo che potrebbe offrire un esperto umano: con la classificazione delle pagine web, con l'ordinamento per rilevanza dei documenti trovati, con l'interattività tra l'utente e il motore stesso, l'algoritmo che sottostà a questi nuovi

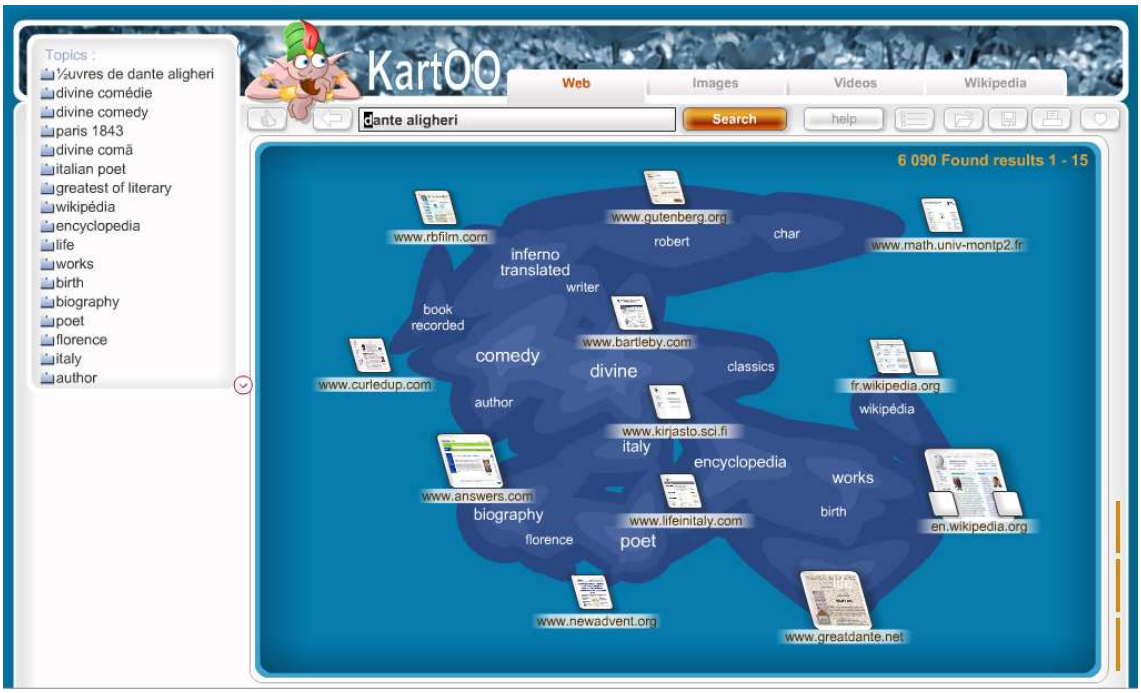

Figura 1.3: Altro esempio di semantic search engine.

motori conduce al più ampio settore dell'intelligenza artificiale.

Proprio grazie a questi algoritmi i nuovi motori consentono di valutare un'enorme quantità di dati e di pagine web, estrapolandone le relazioni, le regolarità, i legami. L'utilità non riguarda solo il settore economico, ma anche quello dell'informazione e della ricerca bibliografica. La ricerca così effettuata nel web apre la possibilità a risultati anche inaspettati su documenti prima quasi introvabili e su legami tra dati e informazioni che possono aiutare il ricercatore con notevole risparmio di tempo e di energie.

Nell' utilizzo dei motori di ricerca di nuova generazione si possono individuare due problemi. Prima di tutto bisogna porre attenzione nella scelta delle pagine web rilevanti e delle classificazioni per argomento proposte dal motore: non sempre il grado di rilevanza è veritiero e non sempre i cluster sono veramente i più adatti per approfondire la ricerca. Non è raro scoprire che alcune delle pagine web presentate sono in realtà irrilevanti, se non addirittura fuorvianti. L'algoritmo, per quanto affinato e preciso, non può sostituire

in toto il contributo della persona studiosa ed esperta di un dato argomento, anche se il motore di ricerca rappresenta un aiuto in più rispetto agli strumenti tradizionali presenti in rete.

Inoltre rappresenta un indubbio problema per l'utente italiano la disponibilità di motori che, per lo più, utilizzano esclusivamente la lingua inglese e che effettuano la loro ricerca a partire da pagine web scritte in inglese. I motori di nuova generazione offrono il meglio delle loro potenzialità proprio se i termini della ricerca vengono inseriti in lingua inglese. In caso contrario, i risultati ottenuti sono limitati e devono essere presi in considerazione con cautela. La funzione di clustering resta comunque un'importante innovazione, qualunque sia la lingua utilizzata dal motore o dall'utente, perchè va a cambiare concettualmente l'idea di ricerca nel web.

La ricerca viene effettuata dai motori anche nel contenuto delle pagine web e questo rappresenta un tentativo di fare ordine e di analizzare la straordinaria mole di dati e documenti presenti in rete, resi spesso introvabili per motivi tecnici legati alla scrittura in codice delle pagine web stesse.

1.3 Feed RSS

Una soluzione interessate al problema dell' "information overload" potrebbe ricercarsi nel rendere reversibile i modi convenzionali grazie ai quali si reperiscono le informazioni. Invece di permettere all'utente di cercare le giuste informazioni, permettiamo alle giuste informazioni di andare verso l'utente. Questo approccio ha richiesto lo sviluppo di software dedicati. La tecnologia RSS, si colloca nell'ambito del news overload, che nn è altro che un aspetto particolare del più generale fenomeno dell'"information overload".

I giornali, le reti TV ,provenienti da ogni parte del mondo, rendono quotidianamente pubbliche centinaia e migliaia di notizie, che spesso fra loro sono parzialmente sovrapposte. Gli articoli spesso trattano della medesima tematica e vengono pubblicati in

giornali differenti e in una precisa dimensione "temporale", ovvero accade spesso che stesse news vengano pubblicate più volte nello stesso brevissimo arco di tempo.

Un feed è usato per fornire agli internauti una serie di contenuti aggiornati di frequente. I distributori del contenuto rendono disponibile il feed e consentono agli utenti di iscriversi. L'aggregazione consiste in un insieme di feeds accessibili simultaneamente, ed è eseguita da un aggregatore Internt. Un aggregatore (in inglese: feed reader) è un programma in grado di effettuare il download di un feed RSS (è sufficiente che l'utente indichi al programma l'URL del feed), effettuarne il parsing e visualizzarne i contenuti in base alle preferenze dell'utente. Spesso i feed reader sono dotati di funzionalità avanzate; ad esempio sono in grado di rilevare automaticamente se il produttore del feed ha effettuato aggiornamenti al feed stesso, effettuandone il download a intervalli di tempo regolari. In questo modo l'utente può essere informato quasi in tempo reale quando un sito è stato aggiornato.

Ci sono molti feed reader in circolazione: alcuni sono applicazioni stand-alone, altri funzionano come plug-in all'interno di altri programmi. Altri sono applicazioni in grado di convertire un feed RSS in una serie di post in formato leggibile dai più popolari news-reader come, ad esempio, Mozilla Thunderbird. Esempi di feed reader stand alone sono, invece, FeedReader, software libero (distribuito con licenza GNU GPL) per piattaforme Microsoft Windows, oppure Liferea, per le piattaforme GNU/Linux.

Nella parte sinistra dell'interfaccia (evidenziata in Fig. 1.4) è mostrato l'elenco dei feed impostati dall'utente; nella parte in alto a destra l'elenco dei singoli contenuti del feed selezionato; nella parte principale l'intero contenuto testuale di un articolo. Alcuni feed reader, come ad esempio RssFeedEater, un programma shareware per Windows, permettono anche di inviare via mail o di scrivere su un blog le informazioni ricevute dai feed.

Ci sono poi lettori di feed con molte funzionalità aggiunte: GreatNews permette di aggregare anche i commenti all'interno del post (sui blog che rendono disponibili i

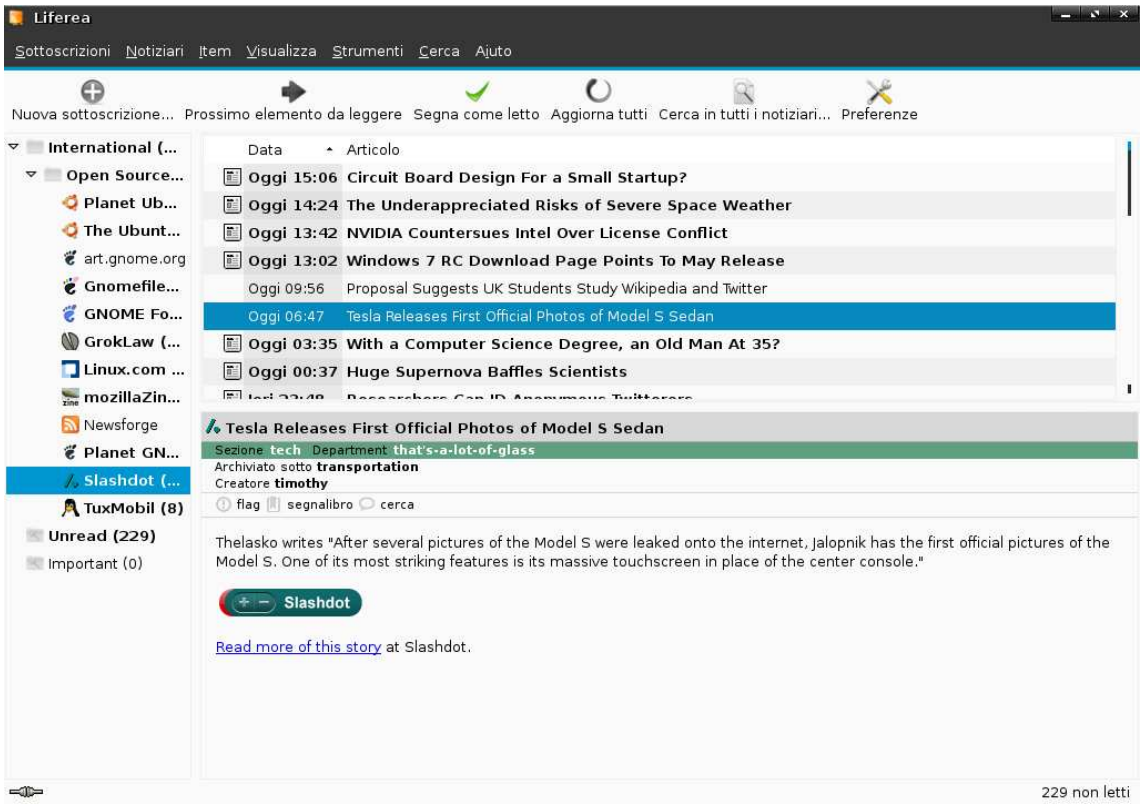

Figura 1.4: Liferea in azione.

commenti via RSS). Uno tra i più grandi aggregatori di notizie in RSS è il sito web Eufeeds che contiene più di mille quotidiani provenienti da tutti gli stati dell'Unione Europea. L'uso principale dei feed RSS (detti anche flussi RSS) attualmente è legato alla possibilità di creare informazioni di qualunque tipo che un utente potrà vedere molto comodamente, con l'aiuto di un lettore apposito, nella stessa pagina, nella stessa finestra, senza dover andare ogni volta nel sito principale. Questo è dovuto al fatto che il formato XML (sul quale si basano i feed) è un formato dinamico.

Il web feed presenta alcuni vantaggi, se paragonato al ricevere contenuti postati frequentemente tramite email:

- Nell'iscrizione ad un feed, gli utenti non rivelano il loro indirizzo di posta elettronica. In questo modo non si espongono alle minacce tipiche dell'email: lo spam, i

virus, il phishing, ed il furto di identità.

- Se gli utenti vogliono interrompere la ricezione di notizie, occorre solo rimuoveer il feed dal loro aggregatore.

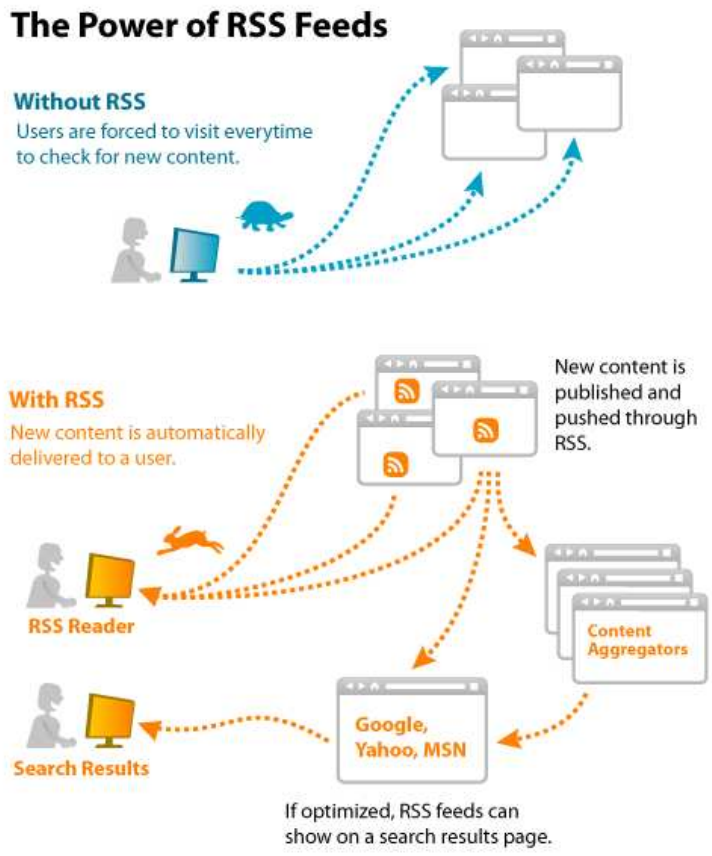

Figura 1.5: Uso di feed RSS.

Senza l'uso dei feed RSS un utente dovrebbe frequentemente visitare il proprio sito di interesse per vedere se in esso si sono verificati o meno aggiornamenti, questo, sostanzialmente, si tradurrebbe in una perdita di tempo che può essere vista come un parente molto stretto dell' "information overload". Con i feed RSS sono le notizie che giungono all'utente e non il viceversa, un esempio è riportato in Fig.1.5

Nonostante i feed RSS aiutono in modo considerevole l'utente in una ricerca, essi, in genere, non garantiscono una soluzione definitiva al problema del "news overload". Infatti questi piccoli software visualizzano news simili che sono state pubblicate su siti diversi, quindi non operano alcun filtraggio oppure alcun raggruppamento particolare; non hanno categorie fissate quindi un utente dovrà manualmente navigare le varte categorie di tutti i giornali digitali di cui ha il feed. Un aspetto importante, lato client, potrebbe essere quello di visualizzare gruppi di news correlate. L'utente potrebbe avere una panoramica generale di un determinato articolo, ma anche l'opportunità di comparare poi fra loro news che trattano la stessa tematica, ma provenienti da fonti differenti.

A questo punto si potrebbero ipotizzare diverse tipologie di feed RSS [6]:

- **Semplici aggregatori**: danno solo un'interfaccia grafica per la visualizzazione di feed RSS da diverse fonti di giornali digitali. Sono previste semplici funzioni che accompagnano l'utente a leggere le news.

- **Nuovi calssificatori**: le news sono classificate in base a criteri la cui scelta, spesso e volentieri,è affidata all'utente .

- **Aggregatori avanzati**: hanno funzioni aggiuntive di classificazione, raggruppamento (clustering) di news.

Sono state implementate diverse strategie per la visualizzazione di news: talune news vengono raggruppate sulla base dei loro contenuti, altre vengono presentate sottoforma di riassunto tralasciando particolari tipici, magari, di un linguaggio specifico quale quello giornalistico. In letteratura, quindi, sono stati presentati vari feed readers che possiamo collocare nella famiglia degli aggregatori avanzati. Fra questi possiamo annoverare Velthune [18], un motore di ricerca di news, che raggruppa, indicizza, classifica news perosnalizzate dall'utente estratte sia dal web che dai news feed, NewsInEssence, che grazie all'utilizzo di particolari algoritmi di clustering riesce ad estrarre un riassunto globale di una data news, e infine RELEVANT^{News} che crea clusters di news simili sulla

base di similarità calcolate sul topic delle news stesse. Di quest'ultimo ci occuperemo nei capitoli successivi.

Capitolo 2

Clustering di news

Quando si parla di tecniche di clustering ci si riferisce a metodi che permettono di partizionare in gruppi che siano omogenei al loro interno e sufficientemente distanti gli uni dagli altri, dato un insieme vasto di oggetti.
Esistono innumerevoli esempi in cui il raggruppamento gioca un ruolo importante. Nell'ambito del "news overload"sono stati progettati numerosi software in grado di clusterizzare le news seguendo numerosi algoritmi applicati a diversi parametri di valutazione.

2.1 Cluster analysis

Il Clustering o analisi dei cluster (dal termine inglese cluster analysis introdotto da Robert Tryon nel 1939 [23]), o analisi di raggruppamento, è un insieme di tecniche di analisi multivariata dei dati volte alla selezione e raggruppamento di elementi omogenei in un insieme di dati.
Il concetto di cluster analysis comprende una serie di algoritmi e metodi per raggruppare gli oggetti di tipo analogo in categorie. Tale analisi si propone di organizzare dei dati in strutture, e di sviluppare tassonomie. In altre parole, l'analisi dei cluster è uno strumento di analisi esplorativa dei dati, che mira a selezionare diversi oggetti in gruppi, in modo

che il grado di associazione tra due oggetti è massima se appartengono allo stesso gruppo e minimo altrimenti.
Tutte le tecniche di clustering si basano sul concetto di distanza tra due elementi. Infatti la bontà delle analisi ottenute dagli algoritmi di clustering dipende essenzialmente da quanto è significativa la metrica, e quindi da come è stata definita la distanza.

La distanza è un concetto fondamentale, dato che gli algoritmi di clustering raggruppano gli elementi a seconda della distanza, e quindi l'appartenenza o meno ad un insieme dipende da quanto l'elemento preso in esame è distante dall'insieme.

Le tecniche di clustering si possono basare principalmente su due filosofie:

- Dal basso verso l'alto (Bottom-Up): Questa filosofia prevede che inizialmente tutti gli elementi siano considerati cluster a sé, e poi l'algoritmo provvede ad unire i cluster più vicini. L'algoritmo continua ad unire elementi al cluster fino ad ottenere un numero prefissato di cluster, oppure fino a che la distanza minima tra i cluster non supera un certo valore.

- Dall'alto verso il basso (Top-Down): All'inizio tutti gli elementi sono un unico cluster, e poi l'algoritmo inizia a dividere il cluster in tanti cluster di dimensioni inferiori. Il criterio che guida la divisione è sempre quello di cercare di ottenere elementi omogenei. L'algoritmo procede fino a che non ha raggiunto un numero prefissato di cluster. Questo approccio è anche detto gerarchico.

Esistono varie classificazioni delle tecniche di clustering comunemente utilizzate. Una prima categorizzazione dipende dalla possibilità che ogni elemento possa o meno essere assegnato a più clusters:

- Clustering esclusivo: in cui ogni elemento può essere assegnato ad esattamente un solo gruppo. I clusters risultanti, quindi, non possono avere elementi in comune. Questo approccio è detto anche *Hard Clustering*.

- Clustering non-esclusivo: in cui un elemento può appartenere a più cluster con gradi di appartenenza diversi. Questo approccio è noto anche con il nome di *Soft Clustering/Overlapping*.

Un'altra suddivisione delle tecniche di clustering tiene conto della tipologia dell'algoritmo utilizzato per dividere lo spazio:

- Clustering Partitivo (detto anche k-clustering): in cui per definire l'appartenenza ad un gruppo viene utilizzata una distanza ed un punto rappresentativo del cluster (centroide, medioide ecc...).

- Clustering Gerarchico: in cui viene creata una visione gerarchica dei cluster, visualizzando i passi di accorpamento/divisione dei gruppi.

- Clustering density-based: in cui il raggruppamento avviene analizzando l'intorno di ogni punto dello spazio. In particolare, viene considerata la densità di punti in un intorno di raggio fissato.

Queste due suddivisioni sono del tutto ortogonali, e molti algoritmi nati come "esclusivi"sono stati in seguito adattati nel caso "non-esclusivo"e viceversa.

Le tecniche di clustering gerarchico producono una rappresentazione gerarchica ad albero.

Questi algoritmi sono a loro volta suddivisi in due classi:

- **Agglomerativo**: Questi algoritmi assumono che inizialmente ogni cluster (foglia) contenga un singolo punto; ad ogni passo, poi, vengono fusi i cluster più "vicini"fino ad ottenere un singolo grande cluster. Questi algoritmi necessitano di misure per valutare la similarità tra clusters, per scegliere la coppia di cluster da fondere ad ogni passo.

- **Divisivo**: Questi algoritmi, invece, partono considerando lo spazio organizzato in un singolo grande cluster contenente tutti i punti, e via via lo dividono in due.

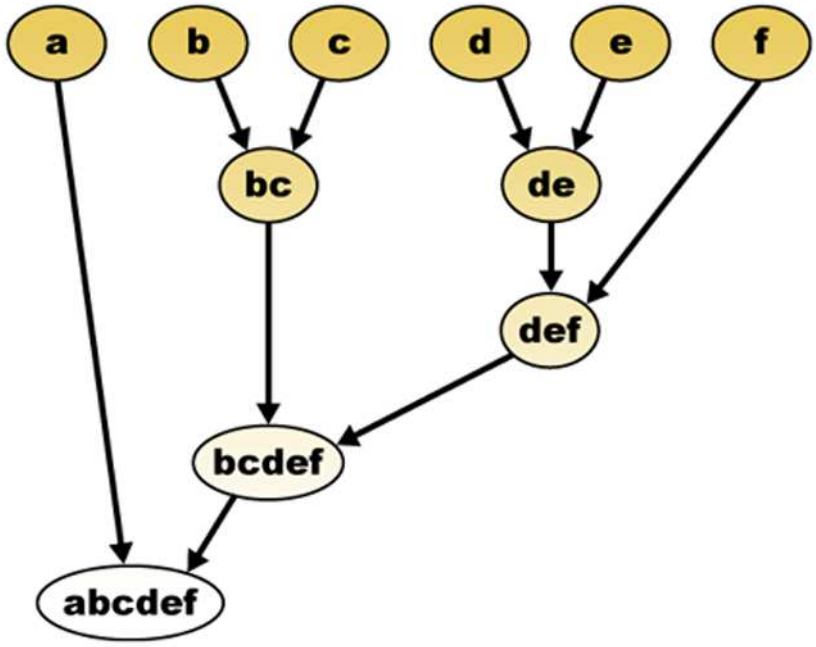

Figura 2.1: Clustering Gerarchico.

Ad ogni passo viene selezionato un cluster in base ad una misura, ed esso viene suddiviso in due cluster più piccoli. Normalmente viene fissato un numero minimo di punti sotto il quale il cluster non viene ulteriormente suddiviso (nel caso estremo questo valore è 1). Questi tipi di algoritmi necessitano di definire una funzione per scegliere il cluster da suddividere.

In entrambe le tipologie di clustering gerarchico sono necessarie funzioni per selezionare la coppia di cluster da fondere (agglomerativo), oppure il cluster da dividere (divisivo).

Nel primo caso, sono necessarie funzioni che misurino la similarità (o, indistintamente, la distanza) tra due cluster, in modo da fondere quelli più simili. Le funzioni utilizzate nel caso agglomerativo sono:

- *Single-link proximity*

  calcola la distanza tra i due cluster come la distanza minima tra elementi apparte-

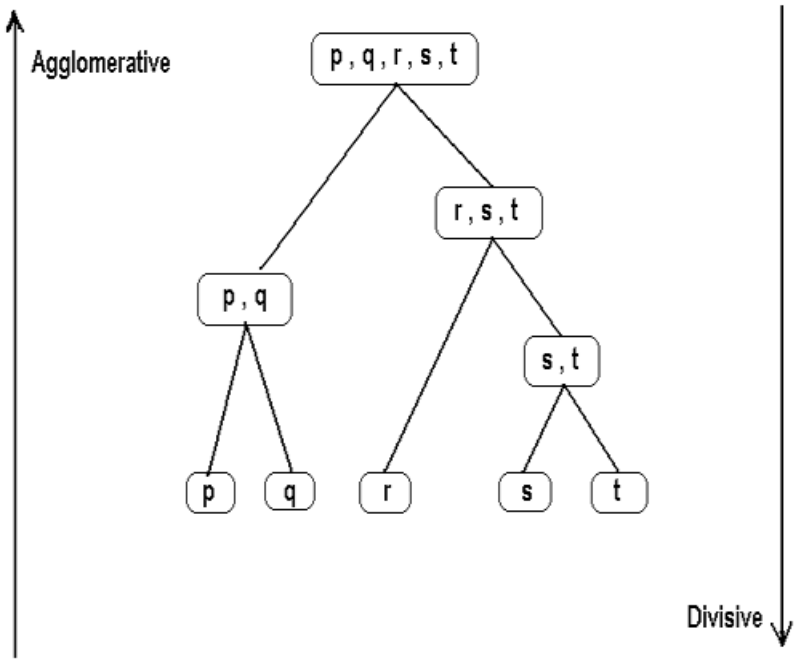

Figura 2.2: Clustering Agglomerativo e Divisivo.

nenti a cluster diversi:

$$D(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y) \tag{2.1}$$

- *Complete-link proximity*

  Questa funzione calcola la distanza tra i due cluster come la distanza massima tra elementi appartenenti ai due clusters:

$$D(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y) \tag{2.2}$$

- *Average-link proximity*

  Questa funzione calcola la distanza tra i due cluster come la media delle distanze tra i singoli elementi:

$$D(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i, y \in C_j} d(x, y) \tag{2.3}$$

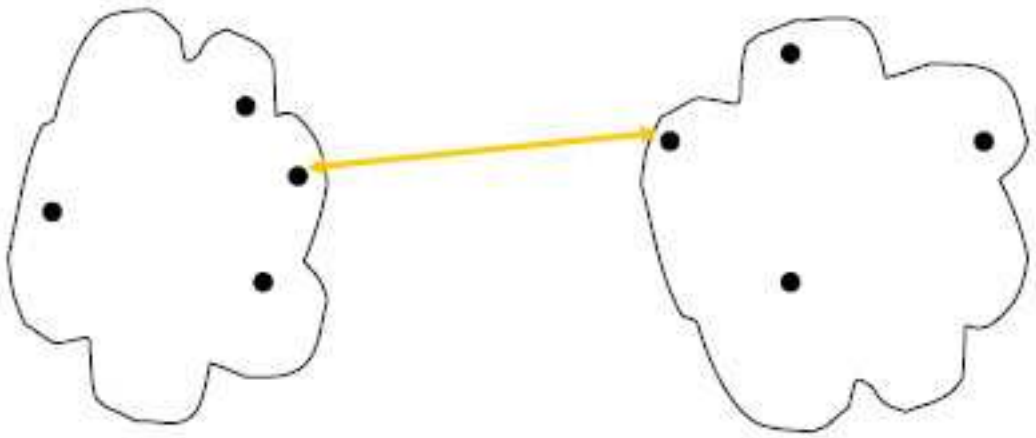

Figura 2.3: Esempio di Single-link proximity.

- *Distanza tra centroidi*

  La distanza tra i due clusters coincide con la distanza calcolata tra i centroidi (o medioidi):

$$D(C_i, C_j) = d(\hat{c}_i, \hat{c}_j) \tag{2.4}$$

Nei quattro casi precedenti, $d(x, y)$ indica una qualsiasi funzione distanza su uno spazio metrico.

Nel clustering divisivo, invece, è necessario individuare il cluster da suddividere in due sottogruppi. Per questa ragione sono necessarie funzioni che misurino la compattezza del cluster, la densità o la sparsità dei punti assegnati ad un cluster. Le funzioni normalmente utilizzate nel caso divisivo sono:

- *Average internal similarity*

  Questa funzione valuta la similarità media tra i documenti interni ad un cluster: più sono tra loro dissimili (valori bassi di similarità), più il cluster è da suddividere in sottogruppi:

$$D(C_i) = \frac{1}{|C_i|(1 - |C_i|)} \sum_{x,y \in C_i, x \neq y} d(x, y) \tag{2.5}$$

- *Maximum internal distance*

  Questa funzione valuta la distanza massima tra due punti interni ad un cluster. Tale valore è noto anche come 'diametro del cluster': più tale valore è basso, più il cluster è compatto:

$$D(C_i) = \max_{x,y \in C_i} d(x, y) \tag{2.6}$$

Oltre agli algoritmi di clustering appena citati, sono stati oggetto di studio altri tipi di algoritmi presentati in letteratura. Uno fra questi propone una clusterizzazione per testi brevi sfruttando le potenzialità di Wikipedia. Dato un testo di un feed item, si creano due query string e le si usano per recuperare gli articoli che in Wikipedia potrebbero soddisfare tali query dall'indice Lucene. Adesso gli articoli recuperati grazie all'indice servono come parametro di clusterizzazione [4]. Risultati sperimentali indicano che questo metodo di rappresentazione ha aumentato l'accuratezza di molti algoritmi di clustering. Esistono altri tipi di classificazioni seguendo diversi parametri come ad esempio la suddivisione fra articoli che descrivono semplicemente degli avvenimenti e quelli che esprimono un'opinione [36] o ancora si esaminano news stabilendo una relazione temporale fra gli eventi [33] e che provvedono a farne un riassunto [3] o il calcolo delle distanze di similarità nello spazio semantico [14]. Si decide di non approfondire le altre tipologie di clustering, poichè nei sistemi software atti alla clusterizzazione vengono precipuamente utilizzate la classa agglomerativa e quella divisiva.

2.2 Software per il clustering di news

Uno degli obiettivi dell' elaborato di tesi è stato quello di classificare i sistemi che permettono il clustering di news, in modo tale da dare una visione generale e globale delle soluzioni che sono state implementate e adottate per far fronte al problema dell' "information overload"e più dettagliatamente dell'"news overload".

Nome	Gruppo ricerca	Anno	Sistema implementato	Sistema gestione
GoogleNews	Google Inc.	2001	aggregatore news	si
PersoNews	univ Tessalonica	X	aggregatore news	si
NewsInEssence	univ Michigan	2005	www.newsinessence.com	si
News@hand	univ Madrid	X	news divise in 8 categorie non è prevista la personalizzazione	si
SemNews	univ Maryland	2006	http://semnews.umbc.edu/	si
SEAN	univ Buffalo& Stony Brook (NY)	X	esamina pagine HTML	no
Newsjunkie	collaborazione fra USA e Israele	2004	aggregatore news	si
OntoMiner	univ dell'Arizona	2005	esamina pagine HTML	no
Flock	Stanford	2005	aggregatore news	si

X = non pervenuto

Tabella 2.1: Classificazione sistemi 1

Prima di passare ad una descrizione più dettagliata dei sistemi citati nelle tabelle **??**, **??**, **??** e **??** ci soffermiamo sul perchè si è deciso di fare una classificazione di questo tipo. Si può notare come per alcuni sistemi la fonte dati per il clustering di news è un file RSS, in altri casi un sito web. Nel caso di sito web occorre analizzare in maniera cadenzata un sito, costruire un albero semantico ed estrarne i dati, questo ovviamente ha un costo computazionale più alto rispetto ai sitemi che utilizzano i feed RSS per il clustering di news. Nel caso dei feed RSS esistono delle tecniche che rendono più veloce e facile la gestione dei documenti, e anche l'uso di ontologie (tra cui WordNet) aiuta nel clustering di news. Le ontologie, a differenza dei semplici database relazionali, permettono infatti di capire quali relazioni possono esserci fra sostantivi, aggettivi e verbi, agevolando il meccanismo di clustering. Si è visto che molti sistemi danno l'opportunità

Nome	Fonte dati	Acquisizione dati	Tecnica nuova	Semantica funzionamento
GoogleNews	RSS	DBMS	si	sintassi
PersoNews	RSS	DBMS	si	sintassi
NewsInEssence	news	diretta	si	sintassi / lessico
News@hand	RSS	diretta	no	sintassi
SemNews	RSS	DBMS	no	sintassi
SEAN	pagine web	diretta	si	sintassi
Newsjunkie	RSS	diretta	si	sintassi
OntoMiner	pagine web	diretta	si	sintassi
Flock	RSS	DBMS	si	sintassi

X = non pervenuto

Tabella 2.2: Classificazione sistemi 2

all'utente di leggere un riassunto della news piuttosto che l'intera news stessa, ma questi metodi devono essere affiancati a delle metodologie di clustering che seguano il naturale evolversi della news, quindi capire il legame temporale che esiste fra una news pubblicata il giono x e la stessa news pubblicata il giorno dopo. Un altro aspetto importante che si nota in alcuni di questi sistemi, è la possibilità che si da all'utente di personalizzare i parametri del clustering delle news; questo aspetto può essere considerato come una sorta di ottimizzazione dei tempi di ricerca per una news, ovvero un utente sa già dove andare a cercare e prelevare l'informazione di suo interesse.

Si passa ora alla descrizione di ogni sistema citato.

Nome	Uso Ontologie	Servizi esterni(API)	Visualiz risul	Possibilità navigazione
GoogleNews	no	no	news	si
PersoNews	ontologia tematica	no	new	si
NewsInEssence	no	agenti sw:newtroll	riassunto	no
News@hand	no	no	riassunto	si
SemNews	OntoSem	API Google	riassunto	si
SEAN	WordNet	no	no	si
Newsjunkie	no	no	si	si
OntoMiner	no	no	si	si
Flock	no	no	si	si

X = non pervenuto

Tabella 2.3: Classificazione sistemi 3

2.2.1 Google News [12]

Google News è un aggregatore automatico di notizie fornito da Google Inc. Google News prevede la ricerca e la scelta di ordinamento dei risultati in base alla data e ora di pubblicazione (da non confondere con la data e l'ora di notizie).

Gli utenti possono richiedere avvisi via e-mail, sui vari argomenti chiave mediante la sottoscrizione ai Google News Aletrs. Le e-mail vengono inviate agli abbonati non appena sono online nuove news che corrispondono a determinate richieste. Le segnalazioni sono disponibili anche tramite feed RSS.

Gli utenti possono personalizzare le sezioni visualizzate, la loro posizione sulla pagina, e il numero di news che possono essere visibili grazie all'uso di un' interfaccia JavaScript. Il servizio è stato integrato con *Google Search History*, dal novembre 2005.
Al suo passaggio dalla fase beta, è stata aggiunta una sezione, Google Search History,

Nome	Parole chiave	Future work
GoogleNews	aggergatori	X
PersoNews	RSS, macchine di auto apprendimento	evitare falsi positivi e falsi negativi
NewsInEssence	news, riassunti	cercare di rispettare il fattore tempo nei riassunti
News@hand	RSS, web semantico, personalizzazione	X
SemNews	ontoligies	migliorare il prototipo
SEAN	HTML, albero semantico	X
Newsjunkie	news, novelty detection	sfruttare Newsjunkie per altri tipi di contenuti come blog e altro
OntoMiner	HTML, albero semantico	cercare di combinare tecniche semantiche e sintattiche
Flock	RSS, clustering	usare altri tipi di algoritmi di clustering oltre a quello gerarchico

X = non pervenuto

Tabella 2.4: Classificazione sistemi 4

che memorizza le queries di ricerca di un utente permettendogli, in un secondo momento, di navigare news che ha letto in passato(tutto ciò ovviamente è possibile se l'utente si è iscritto al servizio GSH).

Come un normalissimo aggregatore, Google utilizza un proprio software per determinare quali news visualizzare dalle fonti online di news che interroga. Il sistema provvederà da solo a interrogare le proprie fonti, l'utente viene, quindi, tagliato fuori da questo tipo di decisione.

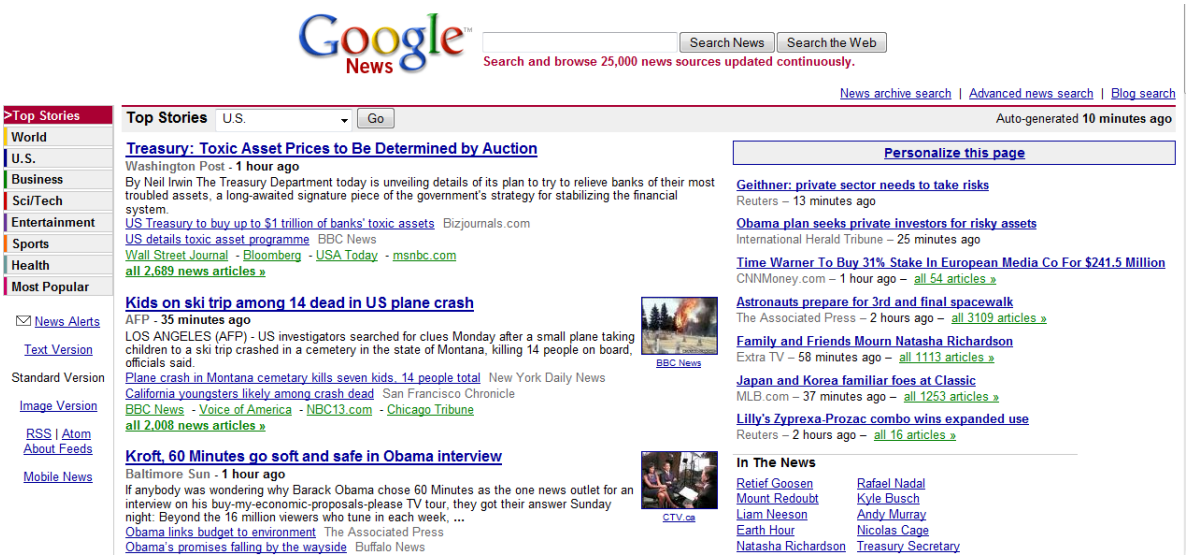

Figura 2.4: Snapshot della home page di GoogleNews.

Da questo punto di vista Google News non è propriamente orientato verso l'utente, poichè potrebbe capitare che ad internauta interessi più una determinata categora di newspaper piuttosto che altri che magari sono inseritoi fra le fonti del sfotware.

La lista delle fonti purtroppo non è conosciuta. In assenza di un elenco vero e proprio molti siti indipendenti, che cercano di emulare il comportamento di Google News, hanno cercato di capire, con le proprie forze, da quali fonti questo famosissimo sfotware prendesse le proprie news.

Come si può vedere dall Fig.2.4 la Home Page di Google News mostra le "Top Stories" seguite dalle varie categorie come "World", "Business"etc.

Per ogni sezione vengono visualizzate le prime tre news appartenenti alla categoria stessa, se l'utente desidera conoscere le news che sono state inserite in una determinata sezione, non dovrà fare altro che cliccare su quella che cattura il proprio interesse; appariranno una serie di "topic"e il numero di news correlate, che possono aggirarsi intorno alle migliaia, e selezionando il topic d'interesse si ha l'opportunità di leggere per intero la news direttamente dalla sua fonte. Google News si avvale dell'appoggio di

circa 4500 fonti, ma come detto precedentemente non esiste una lista precisa, nè tanto meno, i creatori di questo software, hanno reso pubblici gli algoritmi di clustering da loro adottati.

Sulla scia di GoogleNews, sono nati numerosissimi software che in modo trasversale hanno dato, e danno tuttora, il proprio contributo alla riduzione del fenomeno del "news overload"portando l'utente su nuove dimensioni di navigazione delle news.

2.2.2 PersoNews [5]

PersoNews è un lettore di news web-based, supportato da una macchina per l'apprendimento automatico, e filtri semantici, sviluppato dall'universita di Tessalonica (Grecia).

I principali vantaggi di PersoNews sono l'aggregazione di diverse fonti di news, l'apprendimento automatico, la personalizzazione di tutti i feed a cui un utente è iscritto e, infine, la possibilità per ogni utente di vedere più argomenti di interesse utilizzando un semplice modulo di filtraggio semantico.

Dal punto di vista implementativo PersoNews utilizza un classificatore molto semplice basate su tecniche *Naive Bayes* che filtrano le news che sono inutili in quel momento per l'utente.

Il Naive Bayes è un semplice classificatore probabilistico. I classificatori di questo tipo sono basati su modelli di probabilità che incorporano assunzioni di forte indipendenza che spesso non sono riscontrabili nella realtà, per questo motivo vengono detti "naive". In molte applicazioni pratiche, come parametro di valutazione per il modello NB, viene utilizzato il metodo della massima probabilit'a.

L'applicazione web PersoNews ha una doppia funzionalità. In primo luogo, funziona come un normale lettore di RSS, e in secondo luogo, l'utente può scegliere da una ontologia tematica, un argomento di interesse.

In PersoNews, l'utente sceglie un argomento da una gerarchia tematica. Questa funzionalità è di vitale importanza, perché un argomento di interesse può essere condiviso da più fonti di news. Si prenda ad esempio un utente interessato ad un argomento quale il ComputerScience oppure il database. Articoli relativi a questi topic potrebbero apparire in molte fonti. Ad esempio, alcune news possono comparire in un generico giornale di Computer Science, oppure altre possono essere trovate in fonti più specializzate, o ancora l'utilizzo e la descrizione di un database può essere reperita direttamente dai fedd RSS del sito di ORACLE.

In PersoNews tutte queste fonti sono clusterizzate nella stessa categoria, ma un classificatore dedicato, provvederà a personalizzare questo categorie per ogni utente.

Contemporaneamente RSS readers come NewsGator *(www.newsgator.com)* e Google Reader *(www.google.com/reader)* possono aiutare a gestire molti feed RSS ma senza alcun filtraggio semantico, quindi sostanzialmente senza apportare nessuna modifica sostanziale al propblema del "news overload".

Al fine di rafforzare il sistema con un filtro adattativo che utilizza il feedback degli utenti dev'essere implementato un vero e proprio sistema di classificazione con una macchina d'apprendimento automatica.

Perciò il classificatore deve obbedire a queste semplici direttive:

- Dev'essere senza ombra di dubbio un ottimo classificatore per processi di clusterizzazione di testo

- Dev'essere un classificatore incrementale, im modo da mantenere costantemente aggiornate le news quando un utente mando un feedback

- Poichè si ha intenzione di implementare un classificatore per ogni utente e per ogni feed, chiaramente questo classificatore deve avere un costo computazionale estremamente basso.

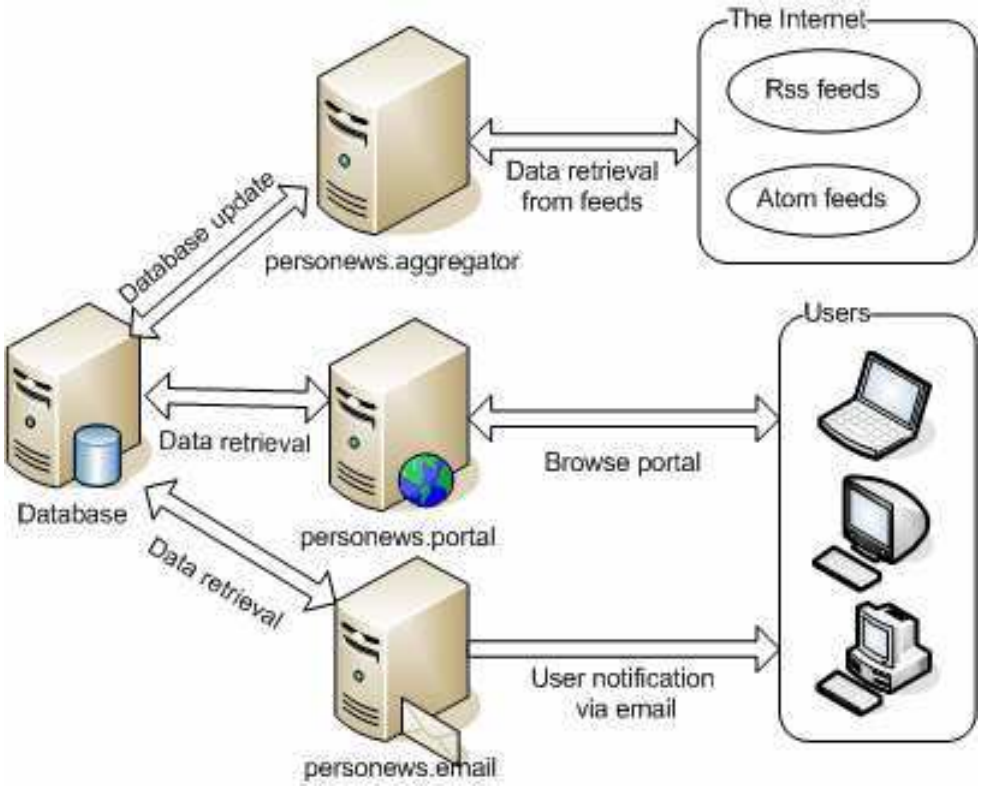

Figura 2.5: Architettura del sistema PersoNews.

- Nelle applicazioni di Text Streaming non esiste una conoscenza a monte di ogni singola parola, e usare un unico vocabolario per un numero smisurato di parole è davvero inefficiente. Quindi occorre un calssificatore che abbia l'abilità di costruire dinamicamente uno spazio di parole non appena arrivano le news.

PersoNews è una applicazione web che fornisce agli utenti la possibilità di controllare una vasta gamma di siti web (feed RSS) e ricevere le notifiche sulle nuove pubblicazioni delle tematiche di loro interesse. Il sistema si compone di tre moduli che funzionano in parallelo usando una banca dati comune per memorizzare informazioni.

I principali moduli sono:

- **Web Site (PersoNews.portal)**: in sostanza è un'applicazione web in cui gli utenti si registrano per poter accedere ai servizi del sistema

- **System update service (PersoNews.aggregator)**: è il processo lato server che organizza gli RSS in modo da scovare nuove pubblicazioni e aggiornare il database di PersoNews. L'aggregatore di PersoNews interroga periodicamente gli URL dei feed, trova le nuove news che sono in giro le processa e le memorizza nel database

- **Email notification service (PersoNews.email)**: è il modo di PersoNews di comunicare con l'utente e avvisarlo degli aggiornamenti dei feed

2.2.3 NewsInEssence [31]

NewsInEssence è un sistema sviluppato dall'Università del Michigan nel 2005, che oltre a clusterizzare le news, come accadeva per il sistema precedentemente descritto, fornisce anche un breve riassunto. Dato uno specifo topic, che può essere una news vera e propria o un insieme di keywords, *NIE* ricerca attraverso la rete le news correlate e le raggruppa formando dei cluster, successivamente il sistema è in grado di generare un riassunto dell'intero cluster.

Siamo di fronte ad un software che tiene conto di due aspetti fondamentali nella guerra al "news overload": **tempo** e **sintesi**.

Per creare un servizio volto alla sintesi di news è necessario considerare il modo in cui queste vengono scritte dai giornalisti.
Molti giornalisti usano una determinata struttura per stilare un articolo, tale struttura è piramidale: una news potrebbe all'inizio dare una visione generale del proprio contenuto, successivamente potrebbe addentrarsi nei particolari ed è proprio seguendo questa struttra che è possibile creare un riassunto.

Ci si chiede adesso con quali tecniche il sistema è in grado di effettuare tali riassunti. La prima idea che sovviene in mente è quella di riuscire, in qualche modo, ad estrarre frasi che per le news prese in considerazione siano abbastanza significative. Tuttavia ogni giornalista, pur seguendo uno standard nella stesura di un articolo, ha un proprio

modo di esprimersi seguendo un proprio stile, e l'apprendimento automatico non potrà mai essere paragonabile al lavoro svolto da un essere umano. Gli altri servizi di news on line presentano all'utente clusters di articoli correlati secondo il topic e quindi gli utenti sono facilitati nella navigazione e nella lettura di news. Tuttavia questi sistemi non prevedono un servizio di sintesi e quindi l'interanuta è costertto a leggere per intero un determinato articolo. NIE provvede a recuparare numerosissime news facendo un excursus prima delle proprie fonti e poi della rete in toto. Il fulcro del sistema NIE è senz'altro il concetto di cluster che tipicamente contiene dalle 2 alle 30 news simili; pero ogni documento presente in un cluster, NIE visualizza il titolo della news, la fonte, la data di pubblicazione e il suo URL.

Nel centro della pagina domina il riassunto del cluster, e appena al di sotto di esso vengono proposti link ad altre sintesi riferite sempre allo stesso cluster. Sulla sinistra appare una barra di navigazione che permette all'utente di navigare altre news, come se fossero un archivio di cluster creati in passato (si faccia riferimento alla Fig.2.6).

NIE crea i suoi cluster in due modi differenti. Data una particolare news, il sistema delega la ricerca di alte news simili a degli agenti software chiamati *NewsTroll* che dapprima seguono degli hyperlinks partendo dalla pagina che contiene la news stessa, alla ricerca di news simili alla data, poi creano dinamicamente una lista di keywords per iterare tale comportamento negli hyperlinks successivi [30]. Le keywords vengono estratte grazie all'uso di una tecnica chiamata *TF*IDF*.

TF*IDF (term frequency inverse document frequency), come indica l'acronimo stesso, calcola il peso di ogni parola all'interno del documento in cui essa è contenuta attraverso delle considerazione sulla sua frequenza nel testo: parole con un TF-IDF elevato implica un forte rapporto con il documento a cui appartengono, questo sta a significare che, se tale termine dovessero apparire in una query, il documento potrebbe essere di particolare interesse per l'utente. L'inverse document frequency si calcola:

$$\mathrm{tf_{i,j}} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2.7}$$

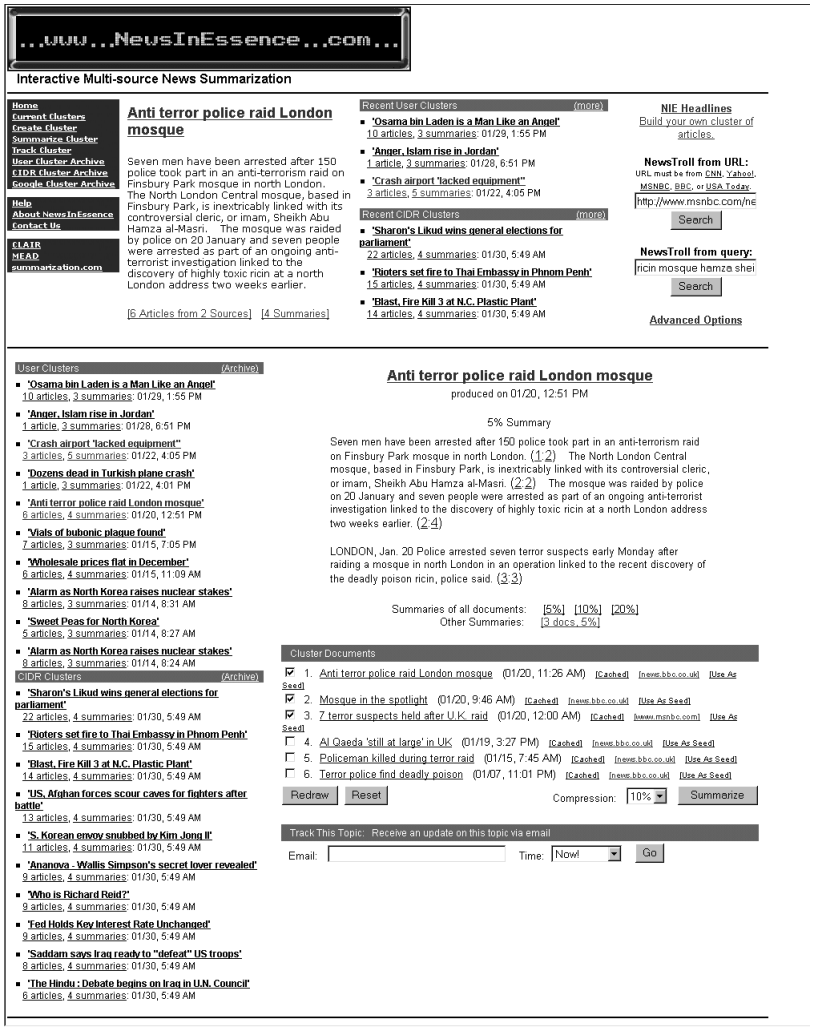

Figura 2.6: NewsInEssence fornt page.

dove ni, j è il numero di occorrenze del termine preso in esame all'interno del dcoumento dj, e il denominatore è la sommatoria delle occorrenze di tutti i termini presenti nel documento. Il documento di frequenza inversa misura quanto pesa un termine nella propria news di appartenenza (si dividono il numero di tutti i documenti per il numero di documenti che contengono il termine, e quindi si calcola il logaritmo di tale quoziente).

$$\mathrm{idf_i} = \log \frac{|D|}{|\{d : t_i \in d\}|} \tag{2.8}$$

con

- $|D|$: numero totali di documenti
- $|\{d : t_i \in d\}|$: numero complessivo di documenti in dui il termine t_i appare (cioè $n_{i,j} \neq 0$).

Allora

$$\mathrm{tfidf_{i,j}} = \mathrm{tf_{i,j}} \times \mathrm{idf_i} \tag{2.9}$$

TF*IDF è tanto alto quanto è alta la ferquenza di un determinata word presente in una news.

Si riporta un piccolo esempio per capire meglio come opera il TF*IDF:

Si consideri un documento contenente 100 parole in cui la parola "computer"appare per 3 volte. In seguito alle formule definite precedentemente, il termine di frequenza (TF) di computer è quindi 0,03 (3 / 100). Ora, supponiamo che vi siano 10 milioni di documenti e la parola computer appare in un migliaio di questi. L'inverse document frequency è calcolato come ln (10 000 000 / 1 000) = 9,21. Quindi il TF-IDF è il prodotto di tali quantitativi: 0,03 * 9,21 = 0,28.

Gli agenti di NIE, usando le keywords create nel primo step, effettuano una ricerca nelle search engine sparse per il web (e qui si esplicita in quali termini NIE prende come fonte la rete in toto)e le news simili trivate vengono inserite nel cluster, le altre invece vengono scartate. La sintesi delle news è affidata a *MEAD* un algoritmo di sintesi di pubblico dominio che utilizza una procedura di clustering nota come *metodo dei centroidi*; MEAD nn fa altro che scovare e prelevare le frasi più importanti dalle news e assemblarle, senza però seguire una struttra ben precisa, non dando perciò all'utente la possibilità di una lettura scorrevole. Sebbene NIE faccia un buon lavoro per la sintesi degli articoli, non bisogna dimenticare che vengono comunque effettuati da software, quindi non saranno mai paragonabili a degli articoli scritti da esseri umani, inoltre il sistema NIE per adesso non prevede di considerare il fattore tempo per la stesura della sintesi di una news;

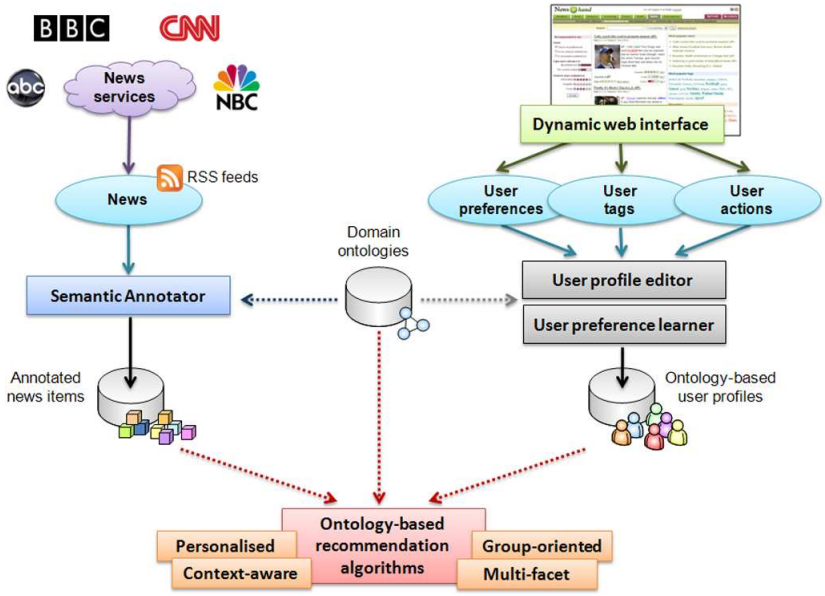

Figura 2.7: Descrizione dell'acquisizione di un profilo di un utente.

un riassunto potrebbe presentare degli avvenimenti scritti in ordine differente dalla loro reale evoluzione.

2.2.4 News@hand [10]

Un altro strumento che sfrutta la tecnologia del web semantico è *News@hand*. News@Hand sfrutta le capacità del web semantico, la tecnologia TF*IDF brevemente descritta nel paragrafo precedente, e possiede inoltre la capacità di creare dei piccoli riassunti delle news.

Tali news, come per altri sistemi, sono periodicamente raccolte e aggiornate grazie all'uso di feed RSS e vengono poi clusterizzate per titolo. Il sistema preso in esame sfrutta la tecnologia AJAX per aggiornare dinamicamente la propria interfaccia grafica che gli permette di immagazzinare e analizzare le news date in input dall'utente e aggiornare le sue preferenze e preparare suggerimenti ad altre news, tutto in tempo reale.

Le news sono classificate in 8 differenti categorie: titoli, mondo, business, tecnologia,

scienza, salute, sport e intrattenimento. Quando un utente non è loggato nel sistema può navigare tutte le categorie ma le news sono visualizzate senza seguire alcun critrio personalizzato, ovvero possono essere visualizzate secondo la data di pubblicazione, la fonte oppure il livello di popolarità.
Se un utente invece è loggato nel sistema, le funzionalità di personalizzazione sono completamente accessbili e l'internauta può navigare le news seguendo le proprie preferenze.

2.2.5 SemNews [28]

SemNews è un servizio di news semantico che, come visto per altri sistemi, gestisce differenti feed RSS e crea un riassunto delle news. SemNews estrae il riassunto dagli RSS e lo processa grazie ad *OntoSem* che è un sistema sofisticato di comprensione del testo. OntoSem prevede non solo l'analisi semantica del testo ma anche quella sintattica.

Come si evince dalla Fig.2.8 questo servizio di clustering di news sfrutta le API di google, ma offre anche un servizio aggiuntivo rispetto agli altri sistemi analizzati: in particolar modo si possono navigare le news per Locazione, ovvero dalla mappa si può scegliere una determinata Nazione e scoprire tutte le news relative a quella Nazione. SemNews punta quindi sull'interfaccia grafica in modo che possa essere più userfriendly possibile.

OntoSem prende input il testo della news ed esegue l'analisi sintattica e semantica per estrarre le frasi più significative. Il preprocessore si occupa di individuare i periodi, espressioni particolari della lingua parlata, il riconoscimento di nomi propri di persona, le date, per quanto riguarda la parte semantica; per la sintattica individua i diversi costrutti grammaticali della frase (si faccia riferimento alla Fig.2.9).

2.2.6 SEAN [34]

Un altro sistema analizzato per il clustering di news è *SEAN*. A differenza dei software e delle applicazioni web precedentemente presentate, SEAN non si focalizza sul

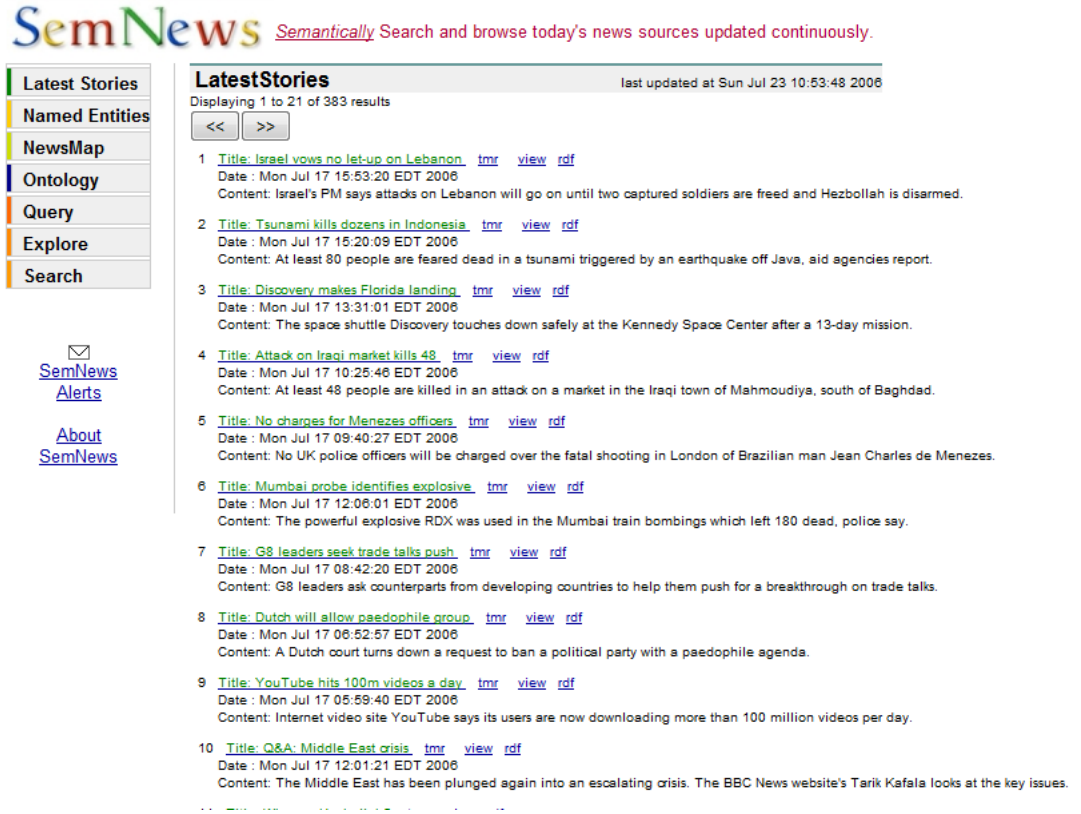

Figura 2.8: SemNews front page.

raggruppamento di nwes tramite feed RSS, ma semplicemente si "limita"a studiare il formato HTML per ricavarne una struttura semantica ad albero, ifatti fra tutti i sistemi visti si avvicina di più a OntoMiner che verrà descritto in seguito. Occorrono due passi per trasformare un documento in formato HTML in una partizione semantica ad albero:

- Identificare i segmenti di un documento che corrispondono a concetti semantici
- Assegnare delle etichette a tali segmenti (informalmente si può affermare che svariati oggetti sono semanticamente correlati se appartengono ad uno stesso concetto).

Come si procede:

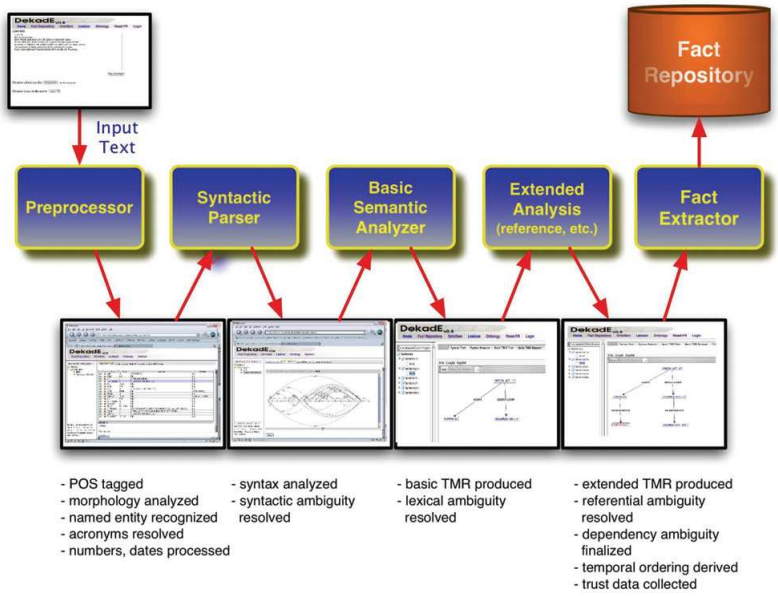

Figura 2.9: SemNews' architecture.

Figura 2.10: New York Times front page.

- Dapprima si vedono dal punto di vista stilistico, cioè osservando proprio la strut-

tura della pagina web, gli oggetti che sono semanticamente correlati fra loro
In Fig.2.10 si possono notare, sulla sinistra, alcune categorie che raggruppano determinati oggetti. Le voci quali "ARTS", "STYLE", "SERVICES", che per questioni di spazio non compare nella sreenshot fornita, non sono altro che tassonomie che raggruppano oggetti che sono semanticamente e logicamente correlati fra loro. Le sottocategorie, come "movies"e "music"sono quindi raggruppate sotto la stessa categoria "madre".

- In secondo luogo si vede come gli oggetti che condividono la stessa natura semantica, condividano anche la posizione all'interno di un documento HTML. Per esempio, scorrendo i vari giornali del web, si può osservare come tali raggruppamenti vengono spesso visualizzati sulla parte sinistra della pagina. La prima idea che viene in mente è quella di associare ad ogni categoria madre, un nodo-radice e per ogni sottocategoria aggiungere, al nodo-radice, un nodo figlio e così via.

SEAN dimostra, quindi, che esiste un forte legame fra l'analisi sintattica e l'analisi semantica di una pagina web. Per l'analisi sematica, e quindi per assegnare le etichette ad ogni segmento SEAN usa *WordNet* [2].

2.2.7 Newsjunkie [17]

Si è più volte detto che identificare le informazioni importanti per un determinato utente è un aspetto importantissimo per la sintesi di testi oppure per la semplice ricerca sul web. Molte tecniche puntano a trovare un insieme di documenti che soddisfano al massimo il bisogno di informazione da parte di un internauta. In questa sezione si presenta una tecnica che serve per identificare nuove informazioni e mostra quali metodi possono essere applicati per gentire il contenuto che evolve costantemente nel tempo.

Si suppone di avere già dei documenti siano clusterizzati o per contenuto o per fonte, e si studiano le differenze intra-gruppi e inter-gruppi. Prendendo in considerazione due

gruppi di news che condividono lo stesso topic, ma non la stessa fonte, possiamo trovare molteplici differenze di opinione e soprattutto di interpretazione degli eventi. (paragone con newsinessence) Ad esempio, si può cercare di esaminare un flusso di news, che hanno lo stesso topic, che evolvono nel tempo, con l'obiettivo di mettere in rilievo gli aggiornamenti veramente importanti e che hanno un contenuto informativo rilevante filtrando così il resto degli articoli che sono superflui.
Newsjunkie crea, come altri sistemi, un riassunto, ma a differenza dei lavori precendenti è in grado di analizzare non intere frasi, ma ogni singola parola. Ovviamente il costo computazionale del sistema aumenta, ma con esso aumenta anche la precisione nella sintesi di news. Si possono individuare tre obiettivi principali:

- Dapprima si descrive un framework che aiuta a tracciare le differenze dei gruppi di news analizzando le parole.

- Successivamente vengono presentati una serie di algoritmi che lavorano sulle news e aiutano l'utente a personalizzarle

- Infine, si descrive un metodo di valutazione che presenta agli utenti un'unica storia madre con le diverse news classificate con metriche di valutazione differenti, che tentano di trovare gli aggiornamenti più importanti alle news e si cerca di capire le reazioni degli internauti a questo tipo di clusterizzazione.

Newsjunkie sfrutta degli algoritmi molto flessibili, infatti il vantaggio principale di questo framework è che essi possono essere applicati, non solo nei newsfeed, ma anche nei blogs e nelle newsgroup.

2.2.8 Estrazione di metadati: OntoMiner [38]

Con la continua crescita del numero di news Web sites e la diversità con la quale i loro contenuti vengono presentati, è stata avvertita una crescente necessità di organizzare

notizie correlate fra loro e di tenerne traccia.
Sono stati sviluppati, a tale scopo, algoritmi in grado di rilevare e utilizzare le normali funzionalità della struttura di un document HTML e trasformarle in strutture semantiche gerarchiche, codificate in formato XML. Il problema di estrarre, gestire e organizzare i dati non strutturati da pagine Web semi strutturate o non strutturate è un problema importante, che ha suscitato l'attenzione di molti ricercatori.

Tali dati devono essere processati e organizzati in maniera uniforme, in modo tale da poter essere successivamente utilizzati per eseguire delle query ad-hoc, oppure per poter estrarre dei riassunti. Esistono una miriade di tecniche che tentano di estrarre informazioni da fonti web semi-strutturate e non strutturate . Fra questi possiamo citare i metodi di apprendimento e di estrazione automatica dei dati che operano su siti Web strutturati per estrarre uno schema e ricostruire il modello delle pagine Web. Come è semplice intuire, questi approcci richiedono che una pagina Wen sia ben strutturata, quindi sono alquanto rigidi. Al fine di sviluppare efficaci tecniche per estrazione di metadati in modo automatico, di solito è più utile concentrarsi in determinato settore di interesse. Uno di questi potrebbe essere quello di giornali on-line e delle news portal, che sono diventati una delle più importanti fonti di aggiornate informazioni.

OntoMiner differisce dalle precedenti metodologie di estrazione dati. E' stato progettato in modo da essere completamente a sè stante, completamente automatico e non necessita dell' intervento umano.
I principali contributi del sistema sono tre e vengono descritti come segue:

- Un algoritmo di partizionamento semantico che logicamente segmenti la pagina e raggruppi e organizzi i contenuti in una pagina HTML.

- Un algoritmo di tassonomia che organizza concetti importanti in una serie di siti Web sovrapposti.

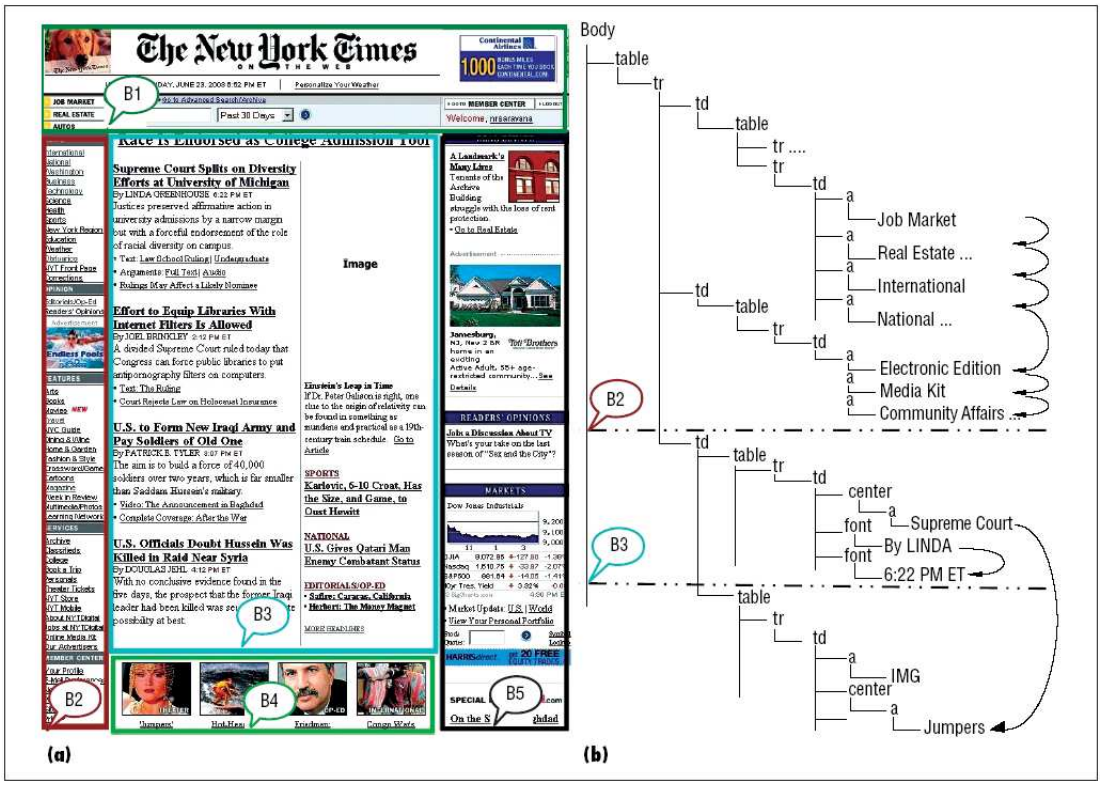

Figura 2.11: Snapshot della home page del New York Times.

- Un algoritmo di estrazione di dati che mette in evidenza istanze individuali con propri attributi dalle pagine Web che appartengono alle medesime categorie.

OntoMiner impiega un algoritmo di partizionameno semantico composto da due fasi: una fase di partizionamento orizzontale (detto Flat) e una seconda fase detta di partizionamento gerarchico (Hierarchical).

L'algoritmo di partizionamento orizzontale *(FP)* individua, all'interno di una pagina Web, vari segmenti logici. Ad esempio, per la homepage del www.nytimes.com, si sono definiti i segmenti logici attraverso delle caselle B1 e B5 in Fig 2.11.a. I confini di segmenti di B2 e B3 corrispondono alle linee punteggiate mostrate nel albero DOM della pagina Web in Fig.2.11.b.

L'algoritmo gerarchico invece calcola relazioni gerarchiche attraverso i nodi dell'albero

DOM in una pagina HTML, dove è contenuto l'intero documento.

Gli algoritmi di clustering che vengono principalmente utilizzati, come già detto in precedenza, sono quelli gerachici.
Tuttavia sono stati implementati e sperimentati anche altri tipi di algoritmi che esaminano il contenuto di un documento, ne vedono la sua evoluzione temporale, la struttura, e creano clusters di documenti basati essenzialmente sul "topic" delle news stesse e usano tali clusters per navigare altre news.

2.2.9 Flock [35]

L'ultimo sistema che rimane da analizzare è *Flock*. Nei sistemi precedenti sono state analizzate diverse tipologie di clustering: il semplice clustering di feeds, si è descritta la possibilità di creare riassunti di news, di personalizzare la clusterizzazione dei feeds. Flock lascia all'utente ampia scelta, cioè sarà l'utente stesso a decidere quali feed RSS dovranno essere clusterizzati.

Flock usa il semplice algoritmo di clustering gerarchico descritto nelle sezioni precedenti. L'algoritmo prende la news selezionata e la inserisce nel cluster "più vicino"; per selezionare il cluster si sfruttano le metriche di similarità quali Matching, Dice, Jaccard, Overlap, Cosine, Information radius e L1 Norm [29].

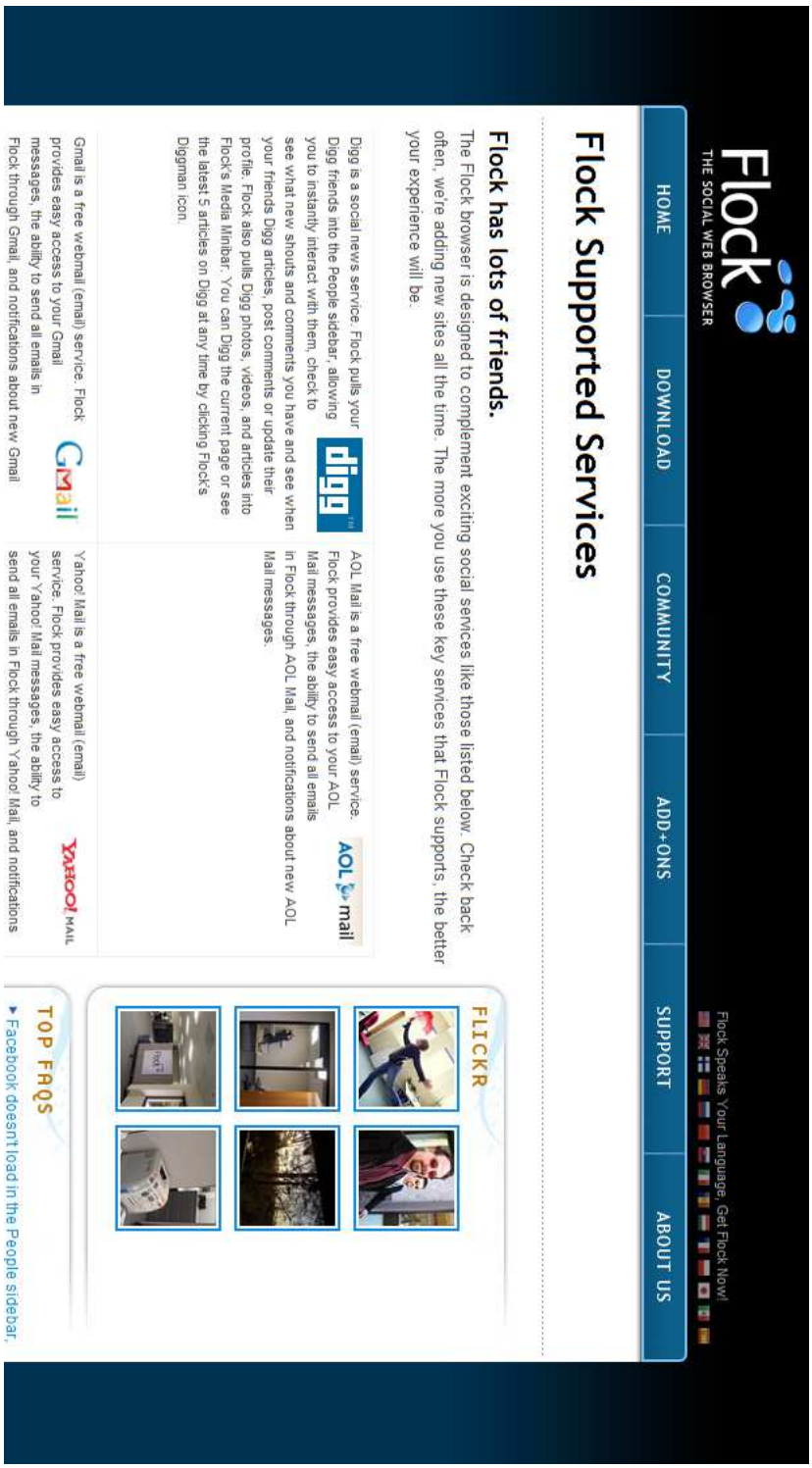

Figura 2.12: Flock's aggregator view.

Capitolo 3

Il Sistema RELEVANT^{News}

In questo capitolo si descriverà il sistema di aggrezione di news RELEVANT^{News} creato dal gruppo DBgroup dell'Università di Modena e Reggio nell'Emilia.

3.1 RELEVANT^{News}

RELEVANT^{News} è un web feed reader che raggruppa automaticamente news accomunate dallo stesso topic pubblicate in newspaper differenti in giorni differenti [7].
Tale strumento è basato su *RELEVANT* un tool sviluppato precedentemente che calcola i "valori rilevanti"come ad esempio un sottoinsieme di valori di una stringa di attributi. Il vantaggio di tale software è che può essere usato o con le sue configurazioni di default, oppure può essere personalizzato dall'utente: l'utente può selezionare diversi parametri per migliorare il sistema di clustering delle news. Si è sperimentato questo software su più di 700 news pubblicate da 30 giornali diversi nell'arco di quattro giorni e i risultati sono stati più che soddisfacenti.

Molti giornali pubblicano le loro news su internet e aggiornano costantemente i loro contenuti, perciò giornalmente vi è un sovraccarico di news pubblicate, di conseguenza esistono centinaia di news che sono parzialmente sovrapposte e che condividono lo stesso

argomento e lo stesso contenuto. Le informazioni pubblicate giornalmente sono talmente tante che un utente non riesce a gestirle tutte efficientemente, d'altro canto gli internauti avvertono l'esigenza di essere aggiornati e informati.

La tecnologia RSS, come ampiamente discusso nei capitoli precedenti, ha fornito un valido aiuto all'utente nel rimanere sempre aggiornato, tuttavia non risolve in maniera significativa il sovraccarico di news, ed è in questo scenario che entra in gioco RELEVANT.
RELEVANT fornisce un metodo che calcola le relazioni sintattiche, lessicale e di dominanza per definire le misure di similarità fra vari attributi.
RELEVANT^{News} è un web feed reader che unisce le capacità di clustering del tool RELEVANT alla potenza di aggiornamento tipica dei feed reader. Applicando RELEVANT ai titoli dei feeds è possibile raggruppare le news pubblicate per la rete e accorparle in clusters sematicamente correlati fra loro.

In particolare ogni cluster contiene delle news simili che seguono determinati parametri:

- **Prospettiva spaziale**: news che condividono lo stesso argomento ma che sono pubblicate in giornali diversi.

- **Prospettiva temporale**: news relative allo stesso argomento ma che sono state pubblicate in momenti diversi.

3.1.1 Il prototipo RELEVANT

RELEVANT si basa sull'idea di analizzare un dominio di attributi e trovare valori che possono essere clusterizzati poichè in forte relazione fra loro. Più formalmente, data una classe C e uno dei suoi attributi At, un **valore rilevante** è definito come:

$$rv^{At} =< rvn^{At}, values^{At} >$$

dove rvn^{At} è il nome del set dei valori rilevanti, mentre $values^{At}$ è l'insieme dei valori veri e propri.

La Fig.3.1 mostra l'architettura funzionale di RELEVANT, tale architettura include i

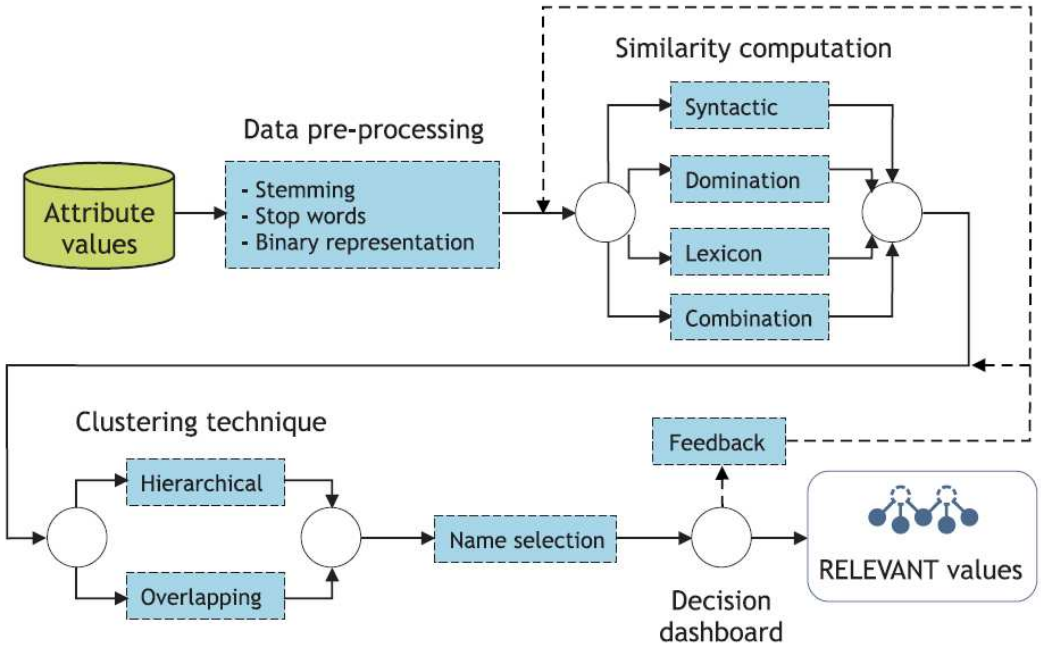

Figura 3.1: Architettura funzionale di RELEVANT.

seguenti processi:

- **Data pre-processing**

  come molti processi di clusterizzazione, il problema principale è quello di trovare valori di attributi e di calcolare funzioni di similarità input per gli algoritmi di clustering. Dopo un processo di stemming (ovvero il processo di riduzione della forma flessa di una parola alla sua forma radice, detta tema), si costruisce una rappresentazione binaria dei valori di attributi, usando tre differenti tipi di misure:

1. *syntactic*

   mappa tutte le parole di valori di attributi in uno spazio astratto, e definisce funzioni di similarità sintattica in tale spazio.

2. *dominance*

   si calcola sugli elementi radice

3. *lexical*

   identifica valori correlati semanticamente ma che sono espressi con una diversa terminologia.

La similarità sintattica è basata sull'idea che le parole che si riferiscono allo stesso oggetto possono avere la stessa etimologia e condividere una stessa radice. Tuttavia, valori sintatticamente simili possono riferirsi a differenti oggetti, di conseguenza un calcolo di similarità basato sul solo metodo sintattico può generare clusters che contengono elementi con significati differenti. Una misura di similarità può essere estratta dalle relazioni di dominanza fra i valori di attributi. Considerando a_1 e a_2, si dice che a_1 domina a_2 se il significato di a_1 è più "generale"di a_2, in termini matematici si definisce la seguente funzione:

$$Contains(X, Y) = true \Leftrightarrow stem(X) \supseteq stem(Y)$$

dove X a Y sono un insieme di parole e *stem* è l'*operatore di stemming*. La dominanza è molto utile per custruire cluster di valori lungo gli *elementi radice*. Infine, per la similarità lessicale, RELEVANT sfrutta WordNet.
In WordNet, le parole inglesi sono raggruppate in insiemi di sinonimi (detti *synsets*). I synsets sono legati da relazioni lessicali oppure sintattico/concettuali e viene data loro una definizione (detta *glossa*). Poichè una parola può essere associata a diversi synsets a causa della polisemia (proprietà che una parola ha di esprimere più significati [25]) ad un utente è chiesto di selezionare manualmente il synset appropriato per ogni termine. D'altro canto, esaminando le similarità lessicali di

WordNet, è possibile raggruppare differenti valori che si riferiscono a synset correlati semanticamente, ovvero due valori diversi che sono presenti in uno o più synsets, sono ptenzialmente simili. Si può quindi calcolare la similarità sulla base di synsets condivisi.

- **Similarity Computation**:
  il calcolo della similarità si divide in due passi; la selezione delle metriche per il calcolare la similarità fra coppie di valori di attributi e la selezione delle misure di similarità che devono essere usate (syntactic, dominance, lexical, o una combinazione delle tre).
  Per quanto riguarda il primo punto, RELEVANT permette all'utente di scegliere le metriche del calcolo di similarità, mettendo a disposizione alcune tecniche usate anche in information retrieval come ad esempio: Simple Matching, Russel & Rao, Tanamoto Coefficient, Sorensen, Jaccard's Similarity. Per quanto riguarda il secondo punto, l'utente a sua discrezione, può fissare determinate soglie per il calcolo della similarità.

- **Clustering technique**:
  questo modulo implementa alcuni algoritmi di clustering per calcolare un insieme di valori rilevanti sulla base del calcolo di similarità selzionato.
  Si può scegliere fra un algoritmo classico di clustering (gerarchico [13], e uno di sovrapposizione [11].

- **Name selection**:
  il nome di un valore rilevante, cioè dato un generico

$$rv_i =< rvn_i, values_i >$$

  rvn_i è il valore più generale di $values_i$
  Il modo più semplice per individuare una lista di rvn_i è quello di usare la funzione

di contenimento. L'utente può scegliere poi il nome più appropriato, fra tutti quelli presentati. Consideriamo ad esempio un valore rilevante che prenda in esame i sostantivi: *assembling operations* e *assembly*. L'algoritmo implementato in RELEVANT sceglierà *assembly* come nome principale poichè è più generico dell'altro elemento e quindi lo contiene.

- **Validation** [8]: RELEVANT implementa un insieme di misure standard per valutare la qualità del cluster:

  - *countRV*:

    rappresenta il numero di valori rilevanti ottenuti. Questo numero dipende dalla soglia fissata dall'utente nella scelta dell'algoritmo di clustering.

  - *avarage, max_elements, variance*:

    rappresentano le statistiche descrittive sul numero di elementi. In particolare, l'*avarage* esprime il numero medio di valori appartenenti ad un valore rilevante,*max_elements* indica la dimensione del cluster più grande e la *variance* mostra il livello di varianza fra le dimensioni dei clusters. Per valori fissati equamente distribuiti nel dominio max_elements è vicino al valore avarage e la variance è più bassa.

  - *count single*:

    numero di valori rilevanti con un singolo elemento. Count single indica un numero basso se l'insieme di valori è equamente distribuito nel dominio.

  - *Rand Statistic index, Jaccard index, Folkes and Mallows index [19]*:

    calcola quanto sono vicini due insiemi di clusters considerando le coppie di valori che appartengono allo stesso cluster in entrambi gli insiemi.

  - *silhouette [32]* (è applicabile solo se viene usato l'algoritmo di clustering gerarchico):

    calcola la larghezza per ogni cluster in base a quanto sono vicini fra loro i vari

elementi presenti in esso. Se il valore silhouette è vicino a 1, allora l'oggetto è stato *ben clusterizzato* ed è stato assegnato al proprio cluster. Se invece tale valore è vicino a -1, allora l'oggetto *non è stato ben clusterizzato*.

- *overlapping degree* (è applicabile solo se si usa un algoritmo di soft clustering): indica la percentuale di elementi che appartengono a più di un valore rilevante.

Da simulazioni effettuate, è risultato che esistono molti cluster con un solo valore; questi clusters non vengono considerati come significativi all'interno del software e vengono detti "outliers".

3.1.2 RELEVANT^{News} architecture

RELEVANT^{News} è una applicazione web che include tre componenti principali:

- un **feed aggregator** che colleziona i feeds selezionati dall'utente.

- un **RSS repository** RELEVANT^{News} ha bisogno di un database per immagazzinare i feeds pubblicati in giorni diversi da differenti testate giornalistiche

- **RELEVANT** che calcola e raggruppa le news simili.

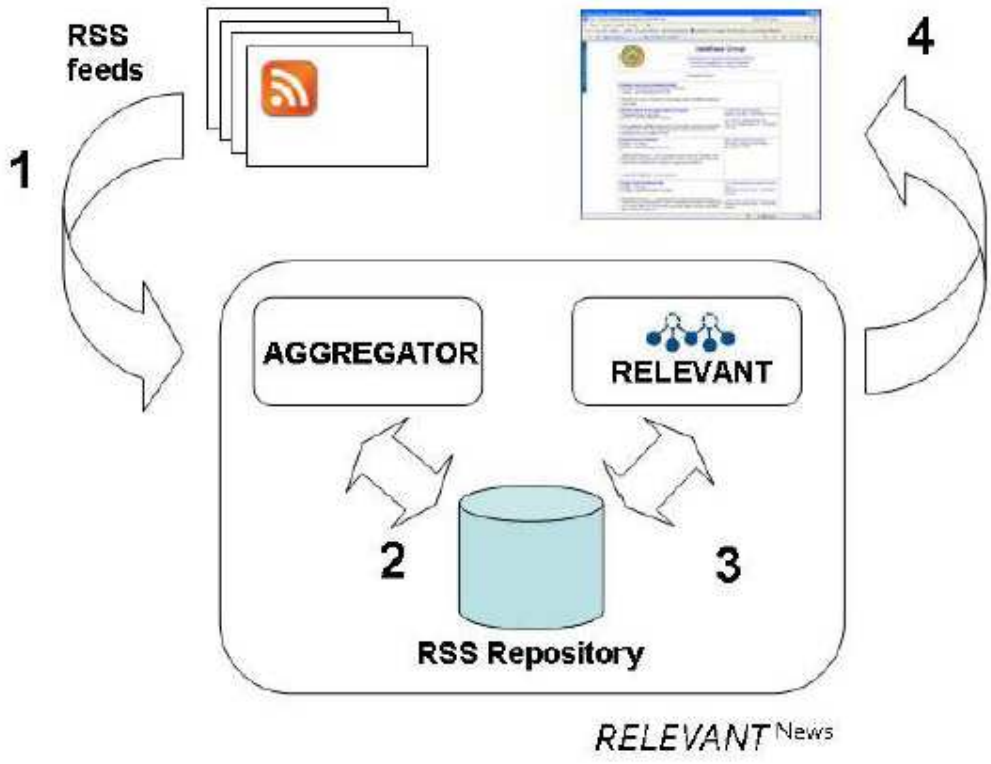

Figura 3.2: The RELEVANT^{News} functional architecture.

L'architettura di RELEVANT^{News} è così composta (si faccia riferimento alla Fig3.2):

1. **selezione dei feeds di news**: è una semplice interfaccia grafica che permette all'utente di selezionare i feeds di news che si reputano essere più interessanti e impostare la politica di aggiornamento dati, ovvero ogni quanto tempo bisogna andare alla ricerca di nuove news.

2. **repository**: è in sostanza un database che memorizza i feeds, quindi rendendo possibile clusterizzare le news per topic prendendo in considerazione anche quelle news che sono state pubblicate tempo prima.

3. **news clustering**: vengono raggruppate le news con gli algoritmi di clustering descritti in precedenza e, per ogni cluster, viene pubblicata una notizia di riferimento. Anche in questo caso esiste un supporto grafico per l'utente, che permette di cambiare diversi parametri creando così cluster ogni volta differenti (un esempio è riportato in Fig:3.3 e in Fig:3.4).

3.2 Matrici Binarie

Nella fase di *Data pre-processing* del software RELEVANT si è parlato di rappresentazione binaria di valori di attributi, usando i tre tipi misure precedentemente descritte. RELEVANT crea, nella fase preliminare, tre matrici binarie [9]:

- *syntactic matching table (MTV)*: è una rappresentazione binaria di tutti i valori di un attributo *At* per l'universo di parole considerato. Ogni riga ha diversi elementi diversi da zero che sono pari al numero di parole contenute nell'attributo associato (cfr.Fig.3.5).

- *root element matching table (MTR)*: mostra gli elementi radice associati ai valori di attributo. Ogni colonna della matrice è un elemento radice, e le righe rappresentano i valori di attributi.

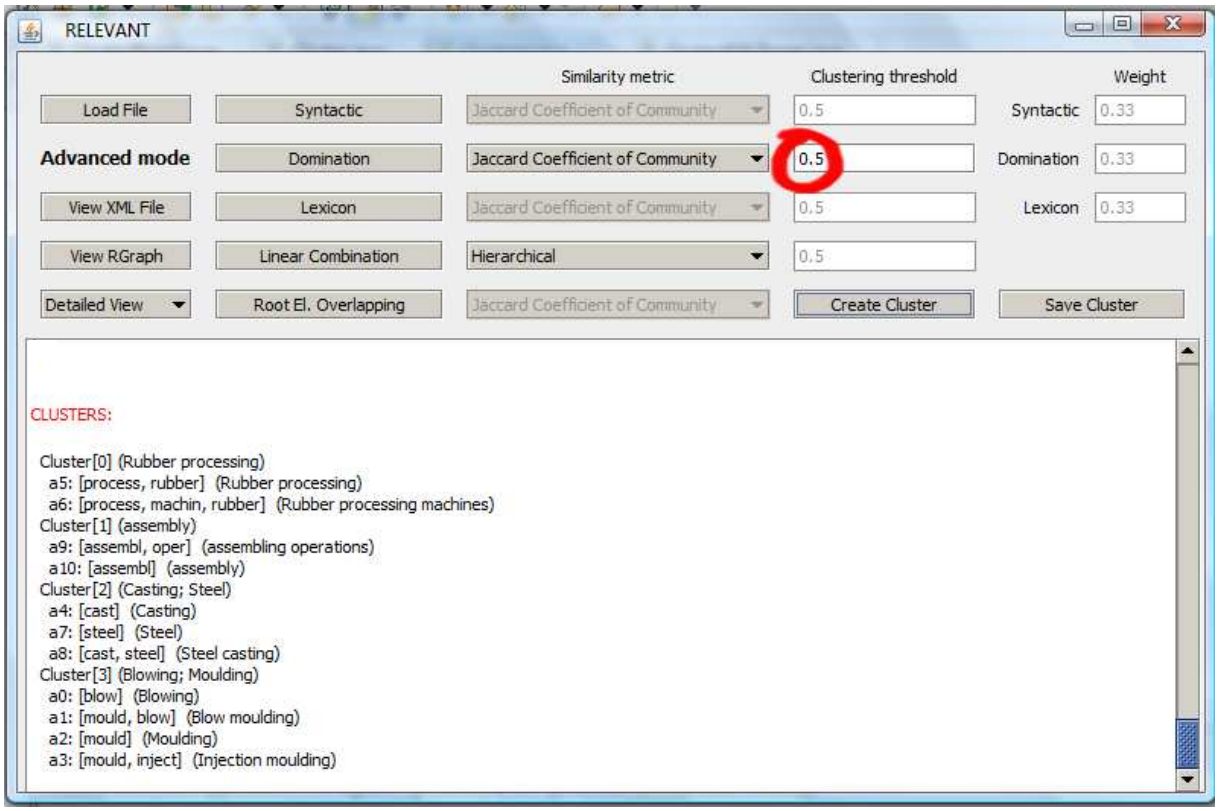

Figura 3.3: Cluster creato con RELEVANT cn soglia 0.5.

- *lexical matching table (MTL)*: mostra i synsets associati ai valori di attributi. Ogni colonna della matrice è un synset, mentre le righe rappresentano i valori di attributo.

Analizzando dal punto di vista matriciale il calcolo della similarità RELEVANT opera seguendo due direzioni: si scelgono le metriche per il calcolo di similarità sulle matrici create nella fase di data preprocessing e si estraggono le matrici di affinità *AMV*, *AMR* e *AML* che derivano rispettivamente dalle matrici *MTV*, *MTR*, *MTL*.

Focalizzando l'attenzione sul secondo punto, le tre matrici sopracitate, esprimono le tre diverse misure di affinità applicando le metriche di similarità sulle matrici *MTV*, *MTR* e *MTL*. RELEVANT procede alla costruzione delle matrici come segue:

- data una matrice *AMV* (*AMR*, *AML*), si ottiene un generico elemento $e_{i,j}$ cal-

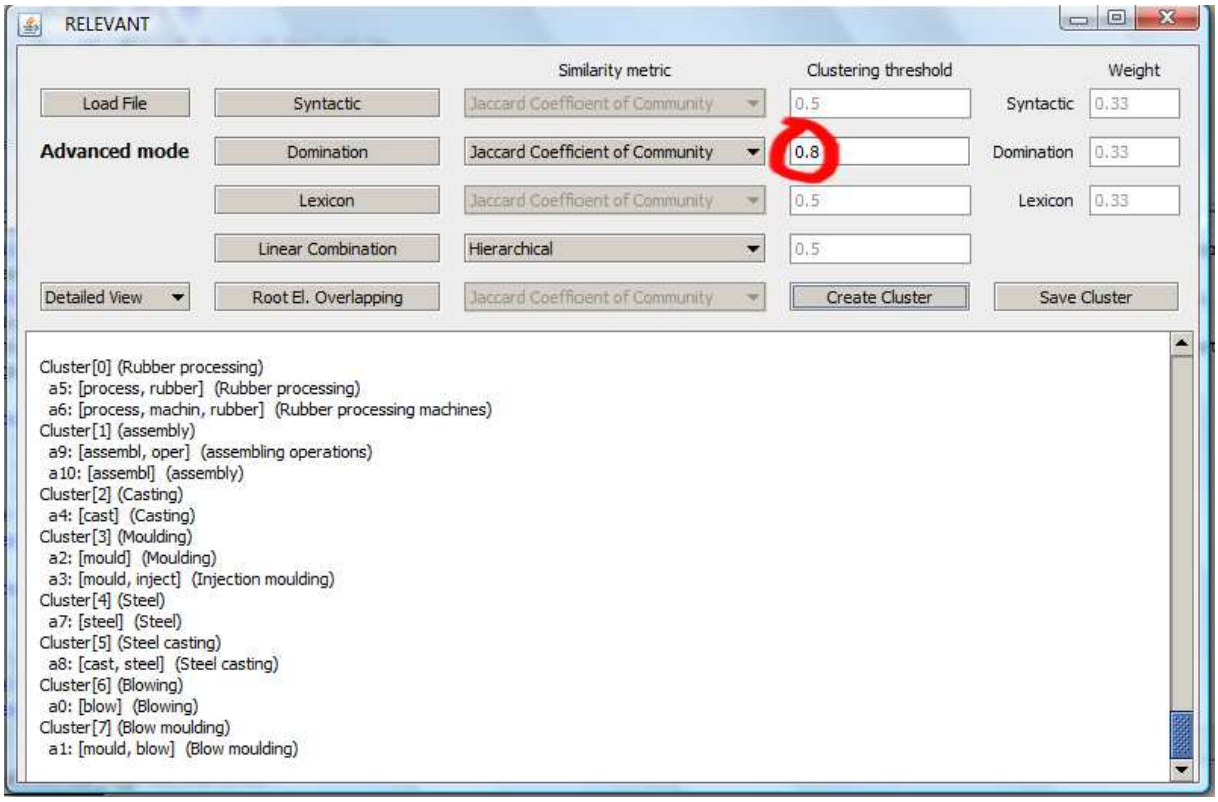

Figura 3.4: Cluster creato con RELEVANT cn soglia 0.8.

MatchingTable MATRIX:

src	u0	u1	u2	u3	u4	u5	u6	u7	u8	u9
a0	0	0	0	0	0	0	0	0	0	1
a1	0	0	0	0	0	1	0	0	0	1
a2	0	0	0	0	0	1	0	0	0	0
a3	0	0	0	0	0	1	0	1	0	0
a4	0	0	0	0	1	0	0	0	0	0
a5	1	0	0	1	0	0	0	0	0	0
a6	1	0	1	1	0	0	0	0	0	0
a7	0	0	0	0	0	0	0	0	1	0
a8	0	0	0	0	1	0	0	0	1	0
a9	0	1	0	0	0	0	1	0	0	0
a10	0	1	0	0	0	0	0	0	0	0

Figura 3.5: Sntactic matching table (MTV) ottenuta per una serie di valori

colando la similarità fra e_i e e_j che sono righe della matrice $MTV(MTR, MTL)$, basata sulle metriche selezionate.

- successivamente RELEVANT combina linearmente AMV, AMR e AML in una

matrice globale di affinità:

$$GAM = ||gam_{hk}||$$

Un elemento è uguale a:

$$gam_{hk} = lc_v \times amv_{hk} + lc_r \times amr_{hk} + lc_l \times aml_h k$$

dove l'utente sceglie i valori di lc_v, lc_r e lc_l tali che $lc_v, lc_r, lc_l \in [0, 1]$ e

$$lc_v + lc_r + lc_l = 1$$

.

La Fig.3.6 mostra la matrice *AMV* calcolata con la metrica *Jaccard* correlata alla *MTV* evidenziata in Fig.3.5

JCC MATRIX:

src	a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
a0	1.0	0.5									
a1	0.5	1.0	0.5	0.333333...							
a2		0.5	1.0	0.5							
a3		0.333333...	0.5	1.0							
a4					1.0				0.5		
a5						1.0	0.666666...				
a6						0.666666...	1.0				
a7								1.0	0.5		
a8					0.5			0.5	1.0		
a9										1.0	0.5
a10										0.5	1.0

Figura 3.6: Matrice di affinità (AMV)

3.3 Il database lessicale WordNet

Si è precedentemente detto che RELEVANT sfrutta WordNet per il calcolo della similarità lessicale. Un'ontologia rappresenta una concettualizzazione condivisa di un certo dominio, e nasce come un accordo consensuale sulla definizione di concetti e relazioni che caratterizzano la conoscenza del dominio stabilito, garantendoci la possibilità di applicare regole d'inferenza (ragionamento) sia per stabilire nuove asserzioni deducibili (nuova

conoscenza sulla base di quella a disposizione), sia per organizzare e recuperare in modo intelligente ed efficiente le informazioni presenti sul web [1]. Essa contiene l'insieme dei concetti (entità, attributi, processi), le definizioni e le relazioni fra concetti, le quali possono essere di vario tipo: tassonomico (IS-A), meronimico (PART-OF) ecc. Dunque è possibile vedere un' ontologia come una rete semantica di concetti appartenenti ad un dominio legati tra loro dalle suddette relazioni. Un' ontologia può presentare vari livelli di formalizzazione, ma deve necessariamente includere un vocabolario di termini (concept names) con associate definizioni (assiomi), e relazioni tassonomiche. Essa è una sorta di "stadio preliminare"di una "base di conoscenza", il cui obiettivo è la descrizione dei concetti necessari a "parlare"di un certo dominio.

Seppur molto evidente per gli esperti di rappresentazione della conoscenza, la differenza tra Ontologie e Database, risulta essere non sempre molto chiara. Molti infatti potrebbero obiettare che entrambi servano ad immagazzinare dati e che l'utilizzo di ontologie nella gestione di grosse quantità di dati non porti nessun valore aggiunti rispetto all'utilizzo di un comune database. Per comprendere a fondo la differenza tra l'utilizzo di un'ontologia e di un database, è necessario comprendere che utilizzare un'ontologia significa adottare un approccio orientato allo *schema concettuale*, che descrive il dominio in cui ci muoviamo, mentre utilizzare un Database Relazionale, significa affrontare il problema con un *approccio orientato ai dati*. Si potrebbe obiettare che anche un database relazionale ha un proprio schema concettuale che descrivere la realtà in cui ci troviamo, ma il potere espressivo dei linguaggi utilizzati per definire i due schemi concettuali (ontologia e database relazionale) è ben differente, e dunque le informazioni che questi ci offrono sono enormemente differenti.
Un' ontologia infatti è molto più ricca di informazioni rispetto ad un semplice schema concettuale di un database. Inoltre, uno schema concettuale di un database, ha lo scopo unico di descrivere le entità (i dati), che andremo ad immagazzinare, mentre un' ontologia ha lo scopo di descrivere i concetti. Dunque, mentre le entità di uno schema

concettuale di un DB sono descritte solamente attraverso una lista di attributi e di relazioni, i concetti di un'ontologia presentano una descrizione molto più ricca. A questo punto, ci si potrebbe chiedere a cosa serve avere un potere espressivo così alto, se un DB relazionale è in grado farci immagazzinare prima e recuperare dopo le informazioni che vogliamo trattare. Un potere espressivo più alto corrisponde ad una migliore organizzazione dei contenuti e dunque ad una migliore fruibilità degli stessi. I contenuti che immagazziniamo sono infatti catalogati secondo lo schema concettuale in uso. La fruibilità dei contenuti stessi dipende ovviamente dal tipo di catalogazione. Dunque più lo schema concettuale che utilizziamo per catalogare i nostri contenuti è elevato, più la catalogazione che possiamo fare è efficiente. Una catalogazione fatta attraverso un semplice DB relazionale, pur se ben progettata, rispecchia in termini di efficienza la poca ricchezza di informazioni che porta con se lo schema concettuale del DB. La catalogazione fatta attraverso uno schema concettuale ricco come quello di un'ontologia, consente di sfruttare in fase di recupero delle risorse catalogate, la ricchezza delle informazioni che l'ontologia stessa ci offre, e dunque la fase di recupero delle risorse catalogate, risulta essere più efficiente.

Un database relazionale infatti è accessibile attraverso query SQL like, ovvero espressioni logiche piuttosto semplici, in grado di farci recuperare solo ciò che abbiamo realmente e fisicamente immagazzinato. L'accesso ad un'ontologia invece, può essere fatto utilizzando dei reasoner, ovvero dei software capaci di ragionare. Attraverso il ragionamento artificiale di questi reasoner, siamo dunque in grado di effettuare ricerche molto più complesse ed efficienti (la complessità delle ricerche dipende come detto prima dalla complessità dello schema concettuale in uso, e dunque dall'ontologia). Un reasoner infatti è in grado di inferire conoscenza, ovvero di apprendere nuova conoscenza sfruttando lo schema concettuale in uso (ontologia) ed i fatti conosciuti al momento della ricerca. Un reasoner è dunque in grado di "emulare"il ragionamento umano, andando a recuperare quelle informazioni non esplicitamente espresse, ma comunque presenti in modo implicito. Ad

esempio, affermiamo che:

1. Mario è padre di Giuseppe
2. Giuseppe è fratello di Andrea
3. Bruno è zio di Andrea

Attraverso l'interrogazione di un semplice DB relazionale non potremmo sapere che Bruno è anche lo zio di Giuseppe. Un reasoner invece, attraverso un processo inferenziale, è in grado di darci questa informazione, che pur se non espressa in modo esplicito, è vera e presente nel nostro dominio. Dunque, in un contesto in cui è necessario immagazzinare e catalogare una grande quantità di informazioni, un approccio ai dati (DB relazionale) risulta essere alla lunga meno efficiente di un approccio concettuale (ontologia), con ovvia conseguenza di una peggiore fruibilità de contenuti, a discapito degli utenti che impegnano il loro tempo e le loro energie per raggiungere risultati non sempre efficienti. In questo contesto si colloca WordNet. WordNet è un lessico semantico per la lingua inglese elaborato dal linguista George Miller presso l'Università di Princeton, che si propone di organizzare, definire e descrivere i concetti espressi dai vocaboli [2].

L'organizzazione del lessico si avvale di raggruppamenti di termini con significato affine, chiamati "synset"(concetto), e sul collegamento dei loro significati attraverso diversi tipi di relazioni chiaramente definite. All'interno dei synset le differenze di significato sono numerate e definite.

Il lessico è disponibile gratuitamente on-line. Nel 2006 il database per la lingua inglese conteneva circa 150.000 parole organizzate in più di 115.000 synsets, la maggior parte dei quali sono connessi a altri synsets da relazioni semantiche.

Le relazioni semantiche variano in funzione del tipo di parola e includono:

- per i sostantivi:
  - iperonimia (hypernyms): Y è un iperonimo di X se ogni X è "una specie di"Y (questo aspetto della semantica viene spesso usato nei motori di ricerca sematici per il clustering di query [21] ;
  - iponimia (hyponyms: Y è un iponimo di X se ogni Y è (una specie di) X;
  - coordinazione: Y è un termine coordinato di X se X e Y hanno un iperonimo in comune;
  - olonimia (holonym):): Y è un olonimo di X se X è parte Y;
  - meronimia (meronym): Y è un meronimo di X se Y è parte X;
- per i verbi:
  - iperonimia (hypernyms): il verbo Y è un iperonimo del verbo X se l'attività X è (una specie di) Y (come viaggio rispetto a movimento);
  - troponimia (troponyms): il verbo Y è un troponimo del verbo X se nel fare l'attività Y si fa anche la X (come mormorare rispetto a parlare);
  - implicazione (entailment): il verbo Y è un'implicazione del verbo X se nel fare X uno deve per forza fare Y (come russare rispetto a dormire);
  - coordinazione: Y è un termine coordinato di X se X e Y hanno un iperonimo in comune.

Capitolo 4

Estensione del software RELEVANT^{News}

In questo capitolo si esporrano le varie soluzioni valutate e adottate per estendere RELEVANT^{News}.
Durante il periodo di attività progettuale, si è deciso di focalizzare l'attenzione sulla creazione di un'interfaccia grafica per rendere più chiara possibile all'utente la distribuzione di news nei clusters e le relazioni che intercorrono fra cluster.

4.1 Scelta dell'interfaccia grafica

Sono state esaminate diverse GUI; in tutte si sono ricercati parametri quali semplicità e chiarezza d'uso. L'idea iniziale è stata quella di cercare un qualcosa che colpisse l'attenzione dell'utente, ponendogli di fronte prima un quadro generale dei cluster creati grazie a RELEVANT e poi con un semplice click, consentirgli di esplorare la news che avrebbe scelto, e tutte quelle correlate appartenenti allo stesso cluster.

Si riportano di seguito alcuni esempi di grafici analizzati:

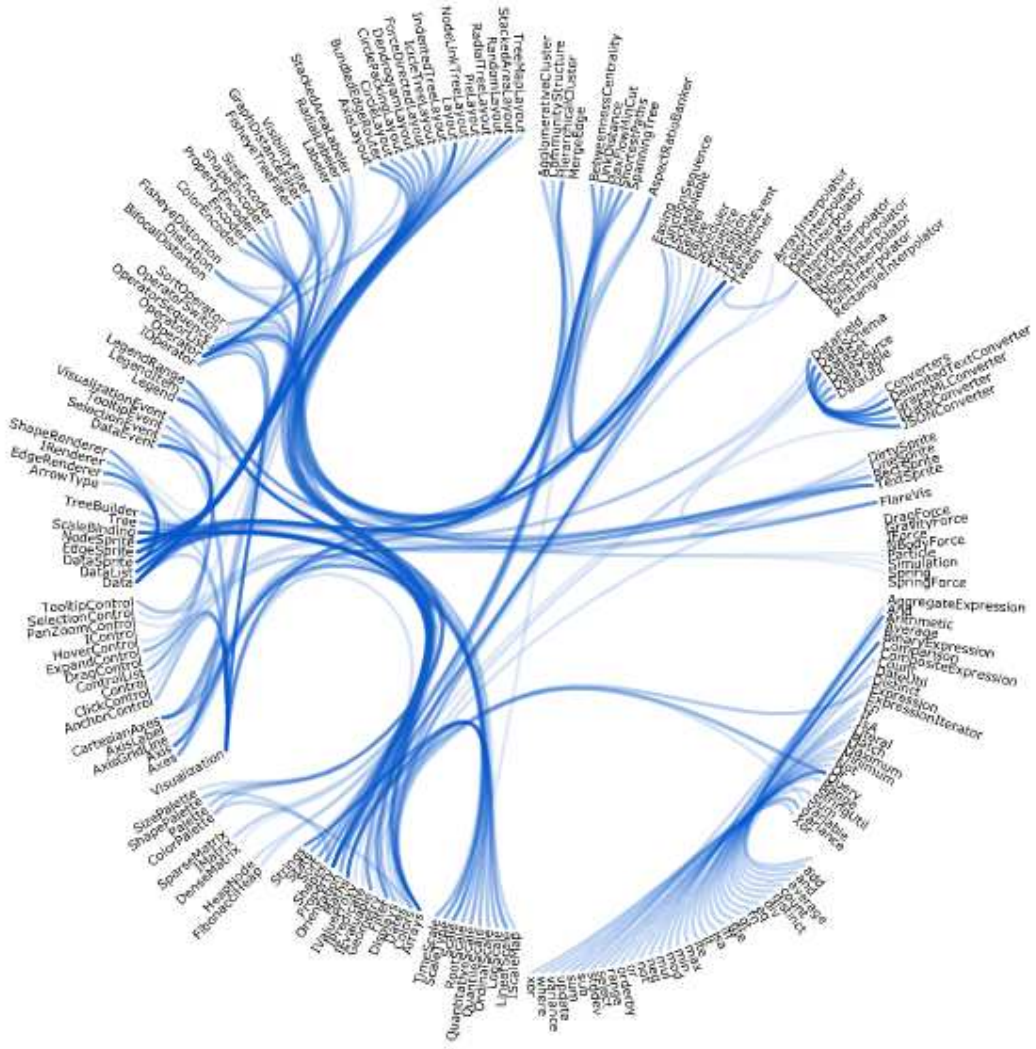

Figura 4.1: Flare Dependency Graph.

Questa visualizzazione mostra le dipendenze tra le classi (java) all'interno della biblioteca Flare [22]. Le classi sono posizionate lungo un cerchio e la lunghezza del raggio indica la profondità della classe nel package. Un link indica che una classe ne importa un'altra.

Quando si muove il mouse su una classe, l'evento metterà in evidenza i collegamenti. Un link rosso mostra tutte le classi importate dalla classe selezionata, mentre il link

verde mostra tutte le classi che importano la classe selezionata (si faccia riferimento alla Fig.4.2).

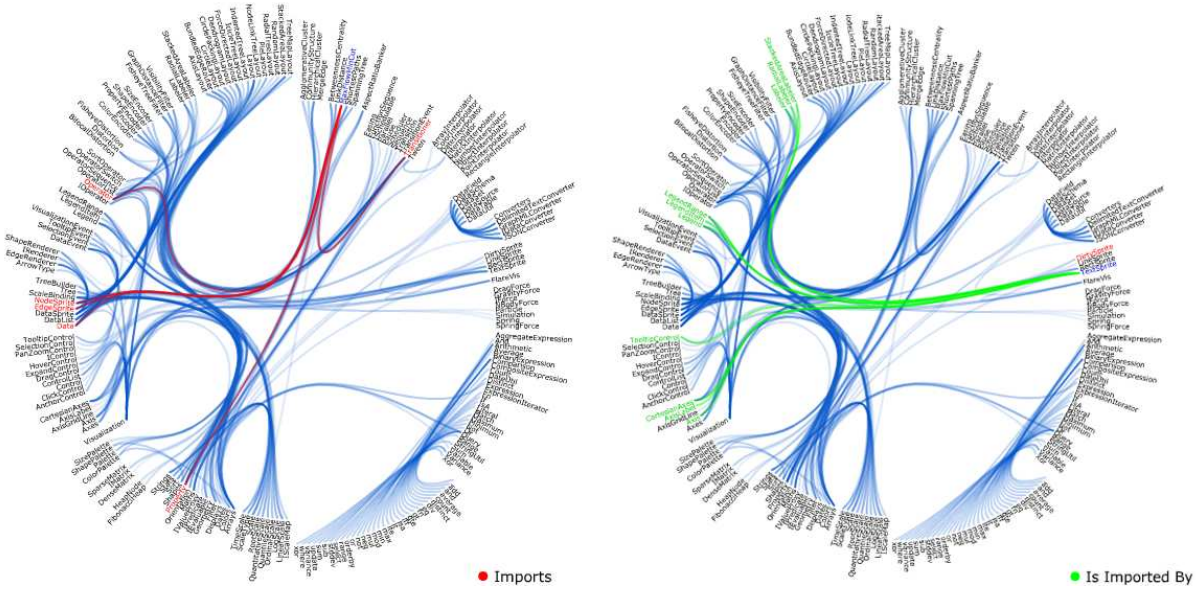

Figura 4.2: Flare Dependency Graph: relazioni fra classi.

Cliccando su una classe il grafo mostra la catena di dipendenze relativa a quella categoria. Questo è calcolata ricorsivamente scorrendo attraverso il grafico per trovare tutte le dipendenze. L'immagine risultante mostra tutte le classi che devono essere presenti per utilizzare la classe selezionata.

Allo stesso modo cliccando una seconda volta il grafo mostra le classi che in qualche modo dipendono dalla classe selezionata. Pensare di usare un tale tool grafico per la visualizzazione dei cluster non è semplice, sarebbe possibile capire le relazioni fra le varie news ma non si riuscirebbe a capire a quale cluster appartengano le varie news.

Altri tool grafici analizzati sono riportati in Fig.4.3 e in Fig.4.4

Tra i due senz'altro *Spectra* [27] è quella che potrebbe avere più impatto su un utente, l'interfaccia grafica è colorita e di facile utilizzo. Ogni colore potrebbe corrispondere ad un Cluster, cliccando sul cluster potrebbero essere visualizzate le news, tuttavia non è stato possible adottare questo tipo di soluzione poichè, essendo un prodotto propri-

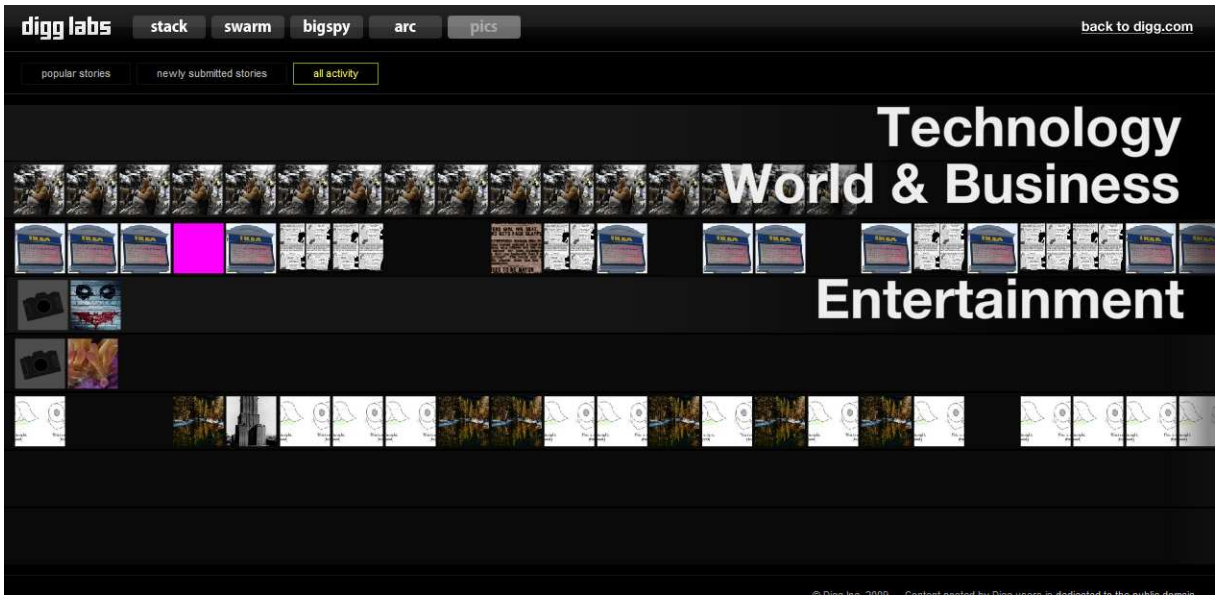

Figura 4.3: digg labs.

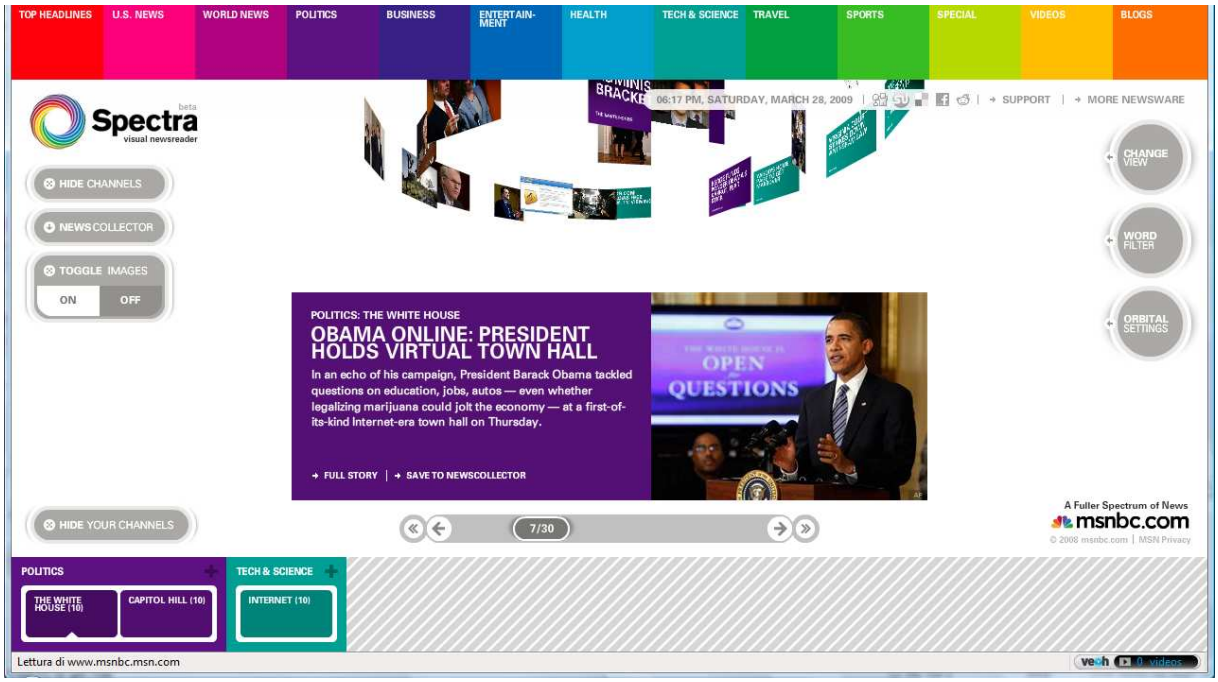

Figura 4.4: Spectra.

etario, non si dispone del codice sorgente, quindi ogni analisi e ogni arrangiamento per RELEVANT è stato praticamente impossibile.

4.2 JavaScript Information Visualization Toolkit (JIT)

Il *JIT* è un avanzato *JavaScript infovis toolkit* basato su cinque papaer che spiegano varie tecniche per la visualizzazione di dati. Il JIT implementa le funzionalità avanzate di visualizzazione delle informazioni, come Treemaps Fig.4.5

Figura 4.5: Treemap.

Spacetree che è riportato in Fig.4.6

e infine un layout di alberi radiali con animazioni avanzate l' RGraph in Fig.4.7.

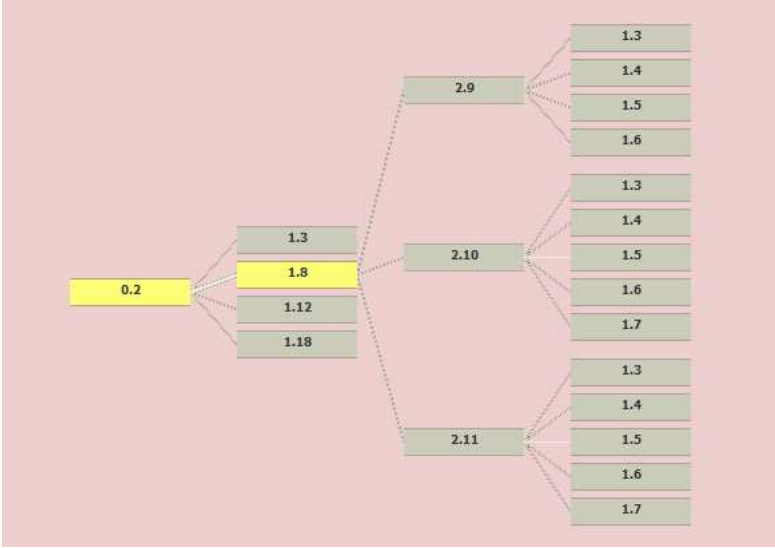

Figura 4.6: Spacetree.

4.2.1 RGraph

RGraph è stato scelto come layout per la visualizzazione di cluster di news per RELEVANT.
Tale layout radiale, mette al centro del grafico il nodo radice e piazza i figli sul primo cerchio concentrico, i suoi nipoti sul secondo cerchio concentrico e così via.
Sono state proposte in letteratura, diverse tecniche di layout radiale.

Queste tecniche variano principalmente su l'angolo di calibrazione (o larghezza angolare), calcolate per un nodo figlio, e uno dei principali paper di riferimento è *Animated Exploration of Dynamic Graphs with Radial Layout* [39], in cui si descrive un modo per animare un layout radiale con transizioni *ease-in* e *ease-out* che permettono transizioni da uno stato ad un altro del grafo facilmente intuibili per l'utente.
La struttura presentata da questo tool grafico è in grado di visualizzare tutti i cluster che RELEVANT crea e, insieme ad essi, tutte le news correlate; permette una visione dapprima globale dei cluster e delle news, successivamente la sola lettura della news di

Figura 4.7: RGraph.

interesse.

4.2.2 Analisi codice Javascript

La possibilità di avere il codice che implementa il layout, ha senz'altro agevolato il compito di adattare la struttura dell'RGraph a RELEVANT, tuttavia l'interazione fra un linguaggio di programmazione quale Java (per RELEVANT) e un linguaggio di scripting ha richiesto diversi studi e diverse prove tecniche.

JavaScript è un linguaggio di scripting orientato agli oggetti comunemente usato nei siti web. Non c'è una vera relazione tra Java e JavaScript; le loro somiglianze sono so-

prattutto nella sintassi (derivata in entrambi i casi dal linguaggio C); le loro semantiche sono piuttosto diverse, e in particolare i loro object model non hanno relazione e sono ampiamente **incompatibili**.

In un primo approccio al problema, si era tentato di "trasformare"il codice javascript in codice Java, ma oltre alla quantità di tempo sprecato, è risultato difficile capire e implementare in classi Java determinate funzioni javascript, come ad esempio il movimento interattivo dei cluster ad ogni click del mouse(Fig.4.8).

```
onClick: function(id) {
    if(this.root != id && !this.busy) {
        this.busy = true;
        this.root = id, that = this;
        this.controller.onBeforeCompute(this.graph.getNode(id));
        var obj = this.getNodeAndParentAngle(id);
        this.tagChildren(obj.parent, id);
        this.parent = obj.parent;
        this.compute('endPos');

        //first constraint
        var thetaDiff = obj.theta - obj.parent.endPos.theta;
        GraphUtil.eachNode(this.graph, function(elem) {
            elem.endPos = elem.endPos.add(new Polar(thetaDiff, 0));
        });

        var mode = (Config.interpolation == 'linear')? 'linear' : 'polar';
        GraphPlot.animate(this, $_.merge(this.controller, {
            hideLabels: true,
            modes: [mode],
            onComplete: function() {
                that.busy = false;
            }
        }));
    }
}
```

Figura 4.8: Cambio posizione dei nodi.

In un secondo momento, poichè la libreria d'interesse è open-source, si è pensato di farla lavorare in locale quindi prendendo il codice sorgente e facendolo comunicare con una pagina HTML. La pagina HTML ha quindi tale struttura:

```
<html>
<head>
<link href= "rgraph-example.css" rel= "stylesheet" type= "text/css"/>
<!--[if IE]>
<script type= "text/javascript" src= "excanvas.js"></script>
<![endif]-->
<script type= "text/javascript" src= "RGraph.js"></script>
<script type= "text/javascript" src= "mootools-1.2.js"></script>
<script type= "text/javascript" src= "example-rgraph.js"></script>
</head>
<body onload= "init();">
<div id= "infovis"></div>
</body>
</html>
```

In questa pagina HTML vengono richiamati i principali file .js (javascript) per poter visualizzare il grafo avviando il proprio browser di default.
Il file *.css* ovvero i Cascading Style Sheets servono per migliorare l'aspetto estetico e al tempo stesso facilitare la creazione o la manutenzione di siti web e questo a prescindere che si tratti di poche pagine o grossi portali.
Se combinati con un linguaggio di scripting, quale per esempio nel nostro caso il JavaScript, danno vita al DHTML ovvero un HTML Dinamico, consentendo di superare quelli che erano considerati un tempo i limiti di html standard.
Con questa tecnica è possibile creare persino delle vere e proprie animazioni sfruttando l'elevata versatilità offerta dal posizionamento degli oggetti sullo schermo, siano essi grafici oppure no. In particolare il file *rgraph-example.css* definisce graficamente la struttra di ogni nodo presente nell'RGraph

```
.node {
        color: white;
        background-color:trasparent;
        cursor:pointer;
        font-weight:verdana;
      font-size: 12px;
        opacity:0.9;
    }
```

Il file *excanvas.js* è la "base di appoggio"sulla quale verrà costruito l'intero RGraph; vengono fissate le dimensioni del canvas, la distanza fra i vari cerchi concentrici, il colore dei nodi, dei rami etc; l'istruzione $< divid = \backslash infovis$" $>< /div >$ permetterà di visualizzarlo.

```
function init() {
     //Set node interpolation to linear (can also be 'polar')
      Config.interpolation =  "linear";
      //Set distance for concentric circles
      Config.levelDistance = 100;
     //Create a new canvas instance.
     var canvas = new Canvas('mycanvas', {
       //Where to inject the canvas. Any div container will do.
       'injectInto':'infovis',
       //width and height for canvas. Default's to 200.
       'width': 900,
       'height': 700,
       //Canvas styles
```

```
        'styles': {
            'fillStyle': '\#ccddee',
            'strokeStyle': '\#772277'
        },
        //Add a background canvas for plotting
        //concentric circles.
        'backgroundCanvas': {
            //Add Canvas styles for the bck canvas.
          'styles': {
                'fillStyle': '\#444',
                'strokeStyle': '\#444'
            }
```

Il file *example-rgraph.js* instanzia un oggetto di tipo RGraph passando come parametro il canvas, e richiama alcune funzioni che servono per: creare l'albero in base all'oggetto json passato come parametro alla funzione, calcolare la posizione dei nodi (ovvero il nodo padre andrà al centro del canvas, i nodi figli sulla prima circonferenza, i nipoti sulla seconda e così via), e infine animare lo spostamento dei nodi ad ogni click del mouse.

```
var rgraph= new RGraph(canvas,  {
        //Add a controller to make the tree move on click.
        onCreateLabel: function(domElement, node) {
          var d = \$(domElement);
          d.set('html', node.name).addEvents(
          {  'click': function() {
              rgraph.onClick(d.id); }
           });
```

```
        }
    }

    //load tree from tree data.
     rgraph.loadTreeFromJSON(json);
     //compute positions
     rgraph.compute();
     //make first plot
     rgraph.plot();
```

4.2.3 L'oggetto JSON

Il codice javascript analizzato nei file RGraph.js ed example-rgraph.js processa un oggetto chiamato JSON, tale oggetto rappresenta la struttura ad albero dei nodi padre e dei nodi figli in modo da permetterne la graficazione in forma radiale.

JSON (JavaScript Object Notation) è un leggero formato di interscambio dati, facile da interpretare e processare. Si basa su un sottoinsieme del linguaggio di programmazione JavaScript, Standard ECMA-262 3a Edizione - dicembre 1999 [24].
JSON è un formato di testo che è completamente indipendenti dalla lingua, ma usa le convenzioni che sono familiari ai programmatori di C, C + +, C#, Java, JavaScript, Perl, Python, e molti altri.
Queste caratteristiche rendono ideale JSON quale valida variabile di interscambio dati. E' stato presentato quale il successore a XML nel browser, JSON aspira ad essere una disposizione di dati semplice e ed elegante per lo scambio di informazioni fra il browser e il server.

L'oggetto JSON è strutturato nel seguente modo:

- Una collezione di coppie: nome / valore.

- Un elenco ordinato di valori.

Si tratta di strutture di dati universale. Virtualmente tutti i moderni linguaggi di programmazione non dovrebbero far fatica a leggere e ad analizzare un oggetto struttrato in questo modo.

Un oggetto JSON si è detto essere un insieme di coppie: nome / valore.
Un oggetto inizia con "{ "e finisce con "}", ogni nome è seguito da ": "e le coppie nome / valore sono separati da una virgola (Fig.4.9) cioè un oggetto possiede un nome che è una proprietà dell'oggetto immaginiamo questo come un nome di variabile normale che è legato al nome dell'oggetto, e l'oggetto possiede il valore di quel nome. Ecco un esempio...

```
var mioPrimoJson = { "nome"    : "Mario",
                     "cognome" : "Rossi",
                     "anni"    : 32 };

document.writeln(myPrimoJSON.nome);    // restituisce Mario
document.writeln(myPrimoJSON.cognome); // restituisce Rossi
document.writeln(myPrimoJSON.anni);    // restituisce 32
```

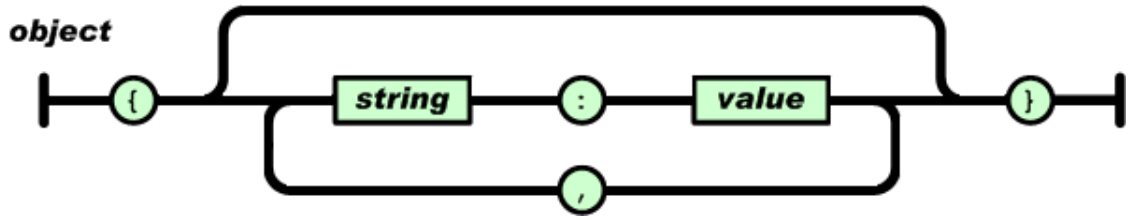

Figura 4.9: Struttura di un oggetto JSON.

Questo oggetto ha 3 proprietà o coppie di nomi/valori. Il nome è una stringa: nell' esempio, nome, cognome e anni. Il valore può essere qualsiasi oggetto di Javascript (e si ricorda che tutto in Javascript è un oggetto quindi il valore può essere una stringa,

numero, array, funzione, persino altri oggetti). In questo esempio i nostri valori sono Mario, Rossi e 32. Mario e Rossi sono stringhe ma gli anni sono rappresentati da un numero e come si può notare questo non rappresenta un problema.

Questa disposizione di dati è appunto denominata JSON in quanto notazione dell'oggetto di Javascript. Ciò che lo rende particolarmente potente è che siccome il valore può essere qualunque tipo di dati, si possono memorizzare altri array ed altri oggetti, annidati secondo necessità. Un array inizia con "["e finisce con "]"e i valori sono separati da una virgola (Fig. 4.10).

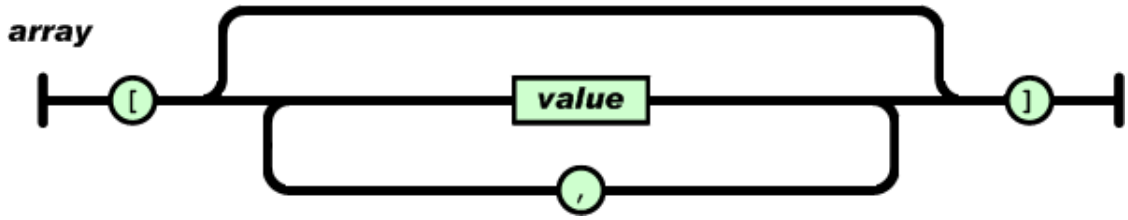

Figura 4.10: Struttura di un array.

Riportiamo un esempio di array JSON:

```
var impiegati ={ "contabili" : [ // "contabili" è un array in "impiegati".
                                 {"nome"    : "Mario", // Primo elemento
                                  "cognome" : "Rossi",
                                  "anni"    : 32 },

                                 {"nome"    : "John", // Secondo elemento
                                  "cognome" : "Smith",
                                  "anni"    : 24 }
                               ], // fine array "contabili".
                  "vendite"   : [ // "vendite" è un altro array in "impiegati".
                                 {"nome"    : "Sally",// Primo elemento
```

```
                                    "cognome" : "Green",
                                    "anni"    : 27 },

                                   {"nome"    : "Jim",  // Secondo elemento
                                    "cognome" : "Galley",
                                    "anni"    : 41 }
                                  ] // fine dell'array "vendite".
                  } // fine "impiegati"
```

Qui "impiegati"è un oggetto. Questo oggetto ha due proprietà o coppie nomi/valori: "contabili"è un array che a sua volta possiede due oggetti di JSON che mostrano i nomi e l'età di due impiegati. Inoltre "vendite"è un altro array che possiede due oggetti di JSON che mostrano il nome e l'età dei due impiegati che lavorano nelle vendite. Tutti questi dati esistono all'interno dell'oggetto "impiegati". Il modo più comune per accedere ai dati di JSON è attraverso la notazione del puntino: semplicemente il nome dell'oggetto seguito da un punto che precede il nome/proprietà a cui vorreste accedere.

```
var mioOggetto = { 'colore' : 'blu' };

document.writeln(mioOggetto.colore); // restituisce "blu".
```

Se il vostro oggetto contiene un altro oggetto allora aggiungere un altro punto e nome

```
var mioOggetto = { 'colore'  : 'blu',
                   'animale' : { 'cane' : 'amichevole' }
                 };

document.writeln(mioOggetto.animale.cane); // restituisce "amichevole"
```

Usando l'esempio degli impiegati qui sopra, se si desidera accedere alla prima persona che ha lavorato nel reparto "contabili":

```
document.writeln(impiegati.amministrativi[0].nome + ' ');
document.writeln(impiegati.amministrativi[0].cognome);
```

Si può anche accedere alla seconda persona che lavora nelle vendite.

```
document.writeln(impiegati.vendite[1].nome + ' ');
document.writeln(impiegati.vendite[1].cognome);
```

Ricapitolando, l'esempio degli impiegati è un oggetto che possiede due array, ciascuno dei quali possiede altri due oggetti. Gli unici limiti alla struttura sono la quantità di spazio disco e di memoria disponibile. Perché JSON può memorizzare oggetti all'interno di oggetti all'interno di oggetti, e degli array all'interno degli array che possono anche memorizzare gli oggetti, non c'è limite virtuale alla quantità di informazioni che JSON può immagazzinare. Data abbastanza memoria e capacità di spazio disco, una struttura di dati semplice di JSON può teoricamente immagazzinare e gestire correttamente grosse quantità di informazioni.

Durante la progettazione dell'interfaccia grafica si è valutata la possibilità di spostare la variabile json, all'inizio inserita nel file RGraph.js, direttamente nella pagina HTML in modo da poter essere vista da tutti i file richiamati all'interno del documento stesso. Il problema sostanzialmente è quello di trasferire la struttura dei cluster nella variabile json, riportarla nel file HTML, e far in modo che RELEVANT (scritto in codice JAVA) scriva in tale file e lo richiami al momento opportuno.

4.3 Interazione JAVA - javascript: creazione e visualizzazione di cluster

Una prima intuizione è stata quella di istanziare un oggetto di tipo *JButton*, nella classe *ClusterJPanel* del package *src.relval*, che, dopo la creazione dei cluster, mi permettesse di visualizzarli nel layout radiale.

```
{   rgraphButton = new JButton();
this.add(rgraphButton, new CellConstraints("3, 9, 1, 1, default, default"));
rgraphButton.setText("View RGraph");
rgraphButton.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent evt) {
    rgraphButtonActionPerformed(evt);
}
});
```

L'evento associato all'rgraphButton è il seguente:

```
private void rgraphButtonActionPerformed(ActionEvent evt) {

    //read from .txt file
    if (this._clusterer!=null){
        String json = this._clusterer.getClusteringToJSON();
           try {
                  FileOutputStream file_html = new FileOutputStream("C:/TESI/prova.html");
                  PrintStream Output = new PrintStream(file_html);
                    Output.println(json);

                } catch (IOException e) {
                  System.out.println("Errore: " + e);
                  System.exit(1);}
                }
     try{

        Process p = Runtime.getRuntime().exec("rundll32 url.dll,FileProtocolHandler C:/TESI/prova.html");
            `    }
                  catch( IOException ioe ){}

}
```

Figura 4.11: RGraph JButton.

Come riportato in Fig. 4.11, la funzione creata, controlla dapprima che vi siano dei cluster, poi richiama un metodo **getClusteringToJSON()** che riporta tali cluster nel formato json.

1.
```
public String getClusteringToJSON(){
      String json = null;
      if (this._groups!=null){
```

```
            json = this.getClusteringToJSON(this._groups);
        }else if (this._clusVis!=null){
            json = this.getClusteringToJSON(this._clusVis);
        }
        return json;
    }
```

2.
```
public String getClusteringToJSON(Cluster[] clusters){
        String json = this.clustersToJSON(clusters);
        return json;
    }
```

Nella funzione (1) si esegue un controllo sull'esistenza dei cluster o gruppi di cluster che possono condividere medesime news, se questi cluster esistono, viene invocata un'altra funzione (2) alla quale vengono passati oggetti di tipo *Cluster[]*, e si passa dunque alla creazione vera a propria della variabile JSON.

Come si evince dalla Fig.4.12, si parte dal nodo radice, che in questo caso abbiamo chiamato "Clusters", e abbiamo dato un nome e un id **univoco**.
E' importante sottolineare tale aspetto, poichè le librerie javascript processano nodi che hanno id diversi; due nodi differenti con uno stesso id, verranno considerati simili e graficamente comparirà un solo nodo.
Ogni cluster ha potenzialmente dei figli, quindi con un ciclo for si scorrono tutti i nodi e per ogni nodo verranno calcolati i rispettivi figli. Questo compito è affidato al metodo: *clusters[g].toJSON(this._termCleaned);*.

Dalla Fig.4.13 oltre alla funzione principale precedentemente menzionata, che prende in input i termini stemmati, si passa ad un altro metodo che ritorna un valore di stringa ovvero *openJSON()*. Tale funzione associa ad ogni cluster "padre"un id ed un nome che assumiamo essere per entrambi la glossa del cluster stesso, inoltre per ogni termine stemmato posseduto dal cluster (quindi logicamente per ogni figlio), viene associata una

```
public String clustersToJSON(Cluster[] clusters){

        String json = "";
        json += "<html>\n";
        json += "<head>\n";

         Codice HTML volontariamente omesso

        json += "<script type=\"text/javascript\">\n";

        if (clusters != null) {
            json += "var json = ";
            json += "{\"id\":\"Clusters\",\n";
            json += "\"name\":\"Clusters\",\n";
            json += "\"children\":[\n ";
            //create json variable
            for (int g = 0; g < clusters.length; g++){
                json += clusters[g].toJSON(this._termCleaned);
            }
            //close parenthesis
            json += "]};";

        }
        else {
        }

        json += "</script>\n";

       Codice HTML volontariamente omesso

        json += "</head>\n";
```

Figura 4.12: Creazione variabile JSON.

chiave *String key = "a"+ j*, e ancora una volta con un ciclo for si associa ad ogni figlio una chiave per l'id ed una glossa per il nome (*objectJSON(key)*).
Tutte le funzioni implementate ritornano delle semplici stringhe che andranno man mano a comporre quella che sarà l'intera variabile JSON.
Ritornando alla Fig.4.11 una volta ottenuta l'intera stringa, l'evento associato al JButton scriverà la pagina HTML e la variabile json in un file (con estensione .html), successivamente verrà invocato un processo che aprirà una pagina del browser di default installato sul sistema operativo permettendo la visualizzazione dei cluster.

Nella Fig.4.14 è riportato un esempio di clusterizzazione di elementi e di visualizzazione degli stessi.
Come si può notare RELEVANT ha creato quattro clusters, che sono stati posizionati

```
public String toJSON(StringCleaning[] termCleaned) {
    varjs = "";
    varjs += this.openJSON();

    for (int j=0; j<termCleaned.length; j++){
        String key = "a" + j;
        if(this.containsKey(key)){
            varjs += this.objectJSON(key);
        }
    }
    varjs += "]},\n";
    return varjs;
}

//clusters
private String openJSON() {
    return (
            "  {\"id\":\""+this.getClusterGloss()+"\",\n" +
            "   \"name\":\"" +this.getClusterGloss()+"\",\n" +
            "   \"children\":[\n" );
}

//object for each cluster
public String objectJSON(String key) {

    String obj_json ="";
    obj_json += "      {\"id\":\"" + key + "\",\n";
    obj_json += "       \"name\":\"" +this.getObjectGloss(key).toString()+ "\",\n";
    obj_json += "       \"children\":[]},\n";

    return obj_json;
}
```

Figura 4.13: Funzioni innestate per il calcolo della var JSON.

sulla prima circonferenza, ogni cluster possiede dei figli che invece sono stati collocati sulla seconda circonferenza, per rispettare la gerarchia.

Supponendo di voler leggere la news *"Injection moulding"* appartenente al cluster *"Blowing;Moulding"*, una volta selezionata, il software la pone al centro del grafo, collocando le altre news sui diversi anelli in modo da mantenere le relazioni di parentela (Fig.4.15).

4.4 Interazione JAVA - javascript: visualizzazione di cluster da file XML

In RELEVANT è possibile inoltre salvare i cluster ottenuti in formato XML.
XML (acronimo di *eXtensible Markup Language*) è un metalinguaggio di markup, ovvero un linguaggio marcatore che definisce un meccanismo sintattico che consente di estendere o controllare il significato di altri linguaggi marcatori [26].
Il passo successivo è quindi quello di riuscire a visualizzare l'organizzazione dei cluster a partire da un file XML precedentemente creato. Anche qui sono stati fatti diversi tentativi prima di approdare alla soluzione finale: uno di questi è stato quello di richiamare il file XML direttamente dal file javascript.

1.
```
function loadXML(xmlfile) {
//Otteniamo un'istanza del parser XML per i diversi tipi di browser
//code for InternetExplorer
if (window.ActiveXObject) {
  xmlDoc=new ActiveXObject("Microsoft.XMLDOM");
}
//code for Mozilla, Firefox, Opera
else
if (document.implementation && document.implementation.createDocument)
{xmlDoc=document.implementation.createDocument("","", null);}
else {
  xmlDoc = false;
}
if(xmlDoc) {
//Impostando async a false ci assicuriamo che lo script
```

```
//attenda il completo
//caricamento del documento XML prima di proseguire l'esecuzione.
xmlDoc.async = false;
xmlDoc.load(xmlfile);
}
return xmlDoc;}
```

2.
```
var xmlDoc;
function Nodi() {
//Crica documento XML
xmlDoc = loadXML('C:/Users/Diletta/Desktop/cluster di prova.xml');
//Puntatore all'elemento radice : <Clusters>
var root = xmlDoc.documentElement;
//Percorro il file salvando le informazioni che mi servono
//Numero dei cluster presenti nel file XML
var nodi = root.getElementsByTagName('Cluster');
//Per ogni nodo presentemi salvo il nome del cluster e il gloss
  var nomeNodi = new Array();
var glossNodi = new Array();
for (i=0; i < nodi.length; i++) {
nomeNodi = nodi[i].getAttribute('name');
glossNodi = nodi[i].getAttribute('gloss');
}
//per ogni cluster
for (i=0; i<nodi.lenght; i++) {
//salvo in un variabile gli oggetti di ogni cluster
var oggetto = root.getElementsByTagName('Object');
//per ogni oggetto salvo le news
```

```
var nomeOggetto = new Array();
for (i=0; i<oggetto; i++){
nomeOggetto = oggetto[i].getAttribute('name');
}
//per ogni nome di oggetto ci salvo il nome dell'elemento
var nomeElemento = new Array();
for (i=0; i < nomeOggetto.length; i++) {
nomeElemento = nodeOggetto[i].childNodes[0].nodeValue;
}}
}
```

Leggere un file XML in remoto con Javascript può essere possibile usando XMLHTTPRequest, esiste tuttavia una valida alternativa che permette anch'essa di leggere file XML (e solo file di questo tipo). Al solito il metodo viene invocato nei due browser piu diffusi, Internet Explorer e Firefox.

La parte di codice (1), ovvero il metodo alternativo, si invoca usando document.implementation per Firefox, mentre per Explorer occorre creare una instanza ActiveX dell'oggetto Microsoft.XMLDOM. I metodi sono entrambi asincroni per default (ma possono essere resi sincroni ponendo la proprietà async a false): xmlDoc.async = false. In particolare nella funzione (1), controlliamo che il documento sia stato caricato.

Nella (2) invece ci accingiamo a leggere i valori del file XML caricato in precedenza. Questo primo approccio tuttavia presenta un problema di fondo, ovvero tener traccia in qualche modo del path del file XML che ci interessa eaminare. Riportando un esempio: si poterbbe pensare di caricare un normalissimo fle .txt, creare i cluster e infine salvarlo, ma a questo punto bisognerebbe in qualche modo memorizzare la path di destinazione del file XML creato e passarla al file .js e non avendo trovato una valida soluzione ci

si è spostati verso altre direzioni. Il software RELEVANT ha implementato una classe (ClusterXmlParser.java) che effettua il parser di un file XML estraendo da esso il numero dei cluster e i suoi oggetti.
Si è pensato quindi di creare un nuovo bottone che permette di visualizzare i cluster a partire da un file XML.

```
{
 graphXML = new JButton();
 this.add(graphXML, new CellConstraints("3, 7, 1, 1, default, default"));
 graphXML.setText("View XML File");
 graphXML.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent evt) {
graphXMLActionPerformed(evt);
}
});
       }
```

Al JButton è ovviamente legato un ActionEvent che è deputato alla creazione della variabile JSON e alla creazione di una **nuova** pagina HTML.
Nella public class implementata innazitutto si crea un oggetto parser: *ClusterXMLParser parser = new ClusterXMLParser()*, si carica il file desiderato e lo si passa al parser: *FileReader fis = new FileReader (fileName); parser.parseXml(fis)*, successivamente si estrae un vettore di cluster invocando il metodo *parser.getClusters()*
Per parserizzare un file XML viene utilizzato il package org.xml.SAX *(Simple API for XML)* che memorizza in un array di vettori i cluster e gli oggetti correlati cn le rispettive chiavi (gli id nella variabile JSON), e le rispettive glosse.
Purtroppo in questa classe le chiavi degli oggetti di ogni cluster vengono salvate in

una stringa di caratteri che appare in questo formato: [a_0, a_1, ..., a_n], quindi occorrerà separarle e assegnarle al giusto cluster.

Come riportato i Fig.4.16 con un ciclo for vengono analizzati i singoli cluster (*clusAfterXML.length*) e per ogni cluster si salva in una stringa la lista di chiavi correlate ai propri oggetti (*String key = clusAfterXML[i].getKeys().toString()*) successivamente sfruttiamo alcuni metodi della classe java STRING, cercando di dividere le chiavi degli oggetti.
Supponendo che la stringa di valori del cluster[0] sia data da [a_0, a_1, a_2], il ciclo while in figura non fa altro che estrarre dalle parentesi i valori di chiave ottenendo così una nuova stringa, ovvero:

$$[a_0, a_1, a_2] \Rightarrow a_0, a_1, a_2$$

A questo punto si è creata una nuova stringa che a sua volta dovrà essere segmentata, dal secondo ciclo while, per estrarre separatamente i valori Tali valori verranno inseriti in un vettore (*Vector vector_key=new Vector()*) che servirà ad assegnare al singolo oggetto di un determinato cluster la chiave corrispondente.

Precedentemente si è detto che questa funzione crea un **nuovo** file html. Tale scelta è motivata dal fatto che un utente potrebbe decidere di vedere nello stesso istante sia i cluster caricati da file XML, sia cluster caricati da semplice file .txt e quindi utilizzando il JButton ViewRGraph.

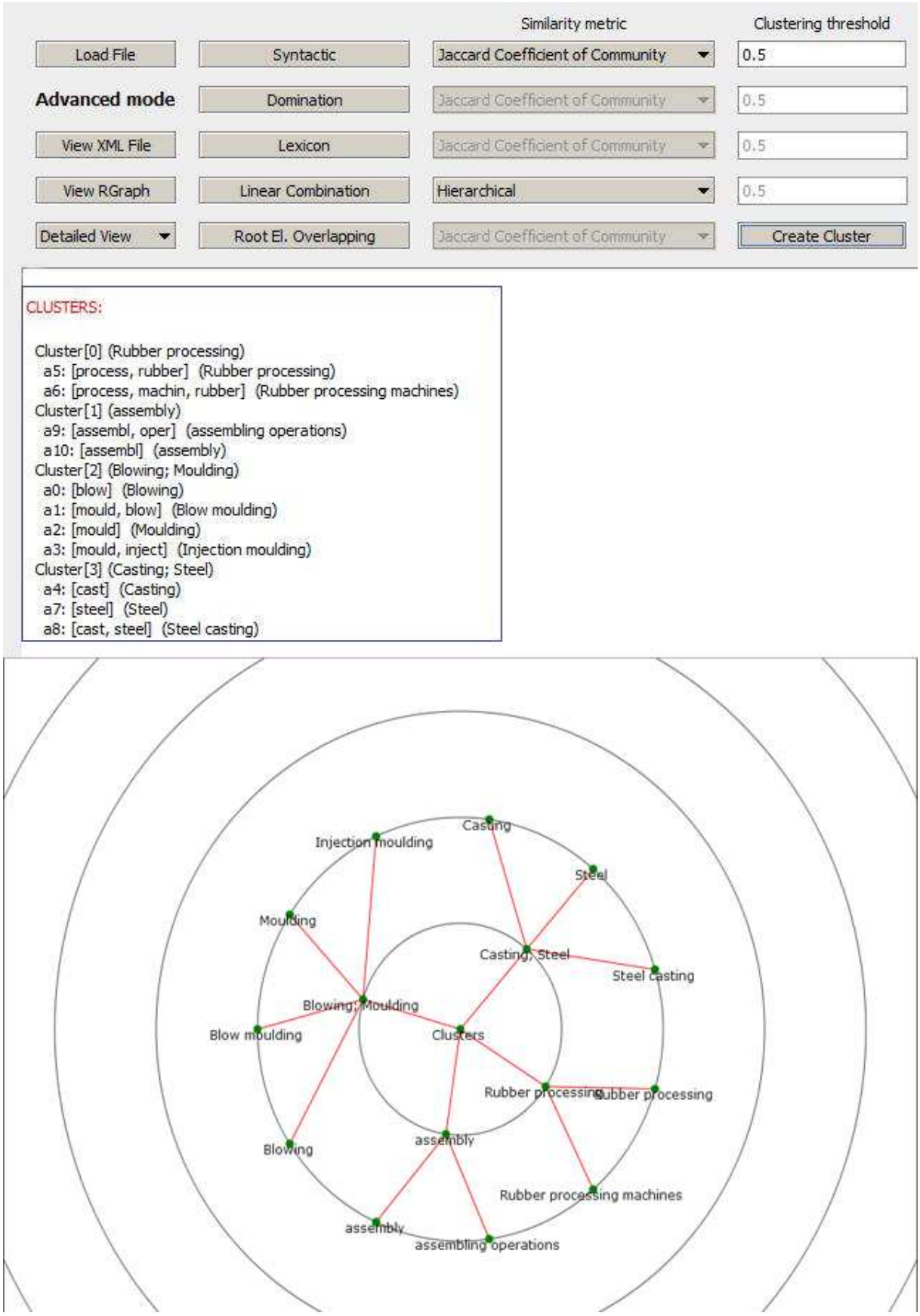

Figura 4.14: RELEVANT & GUI.

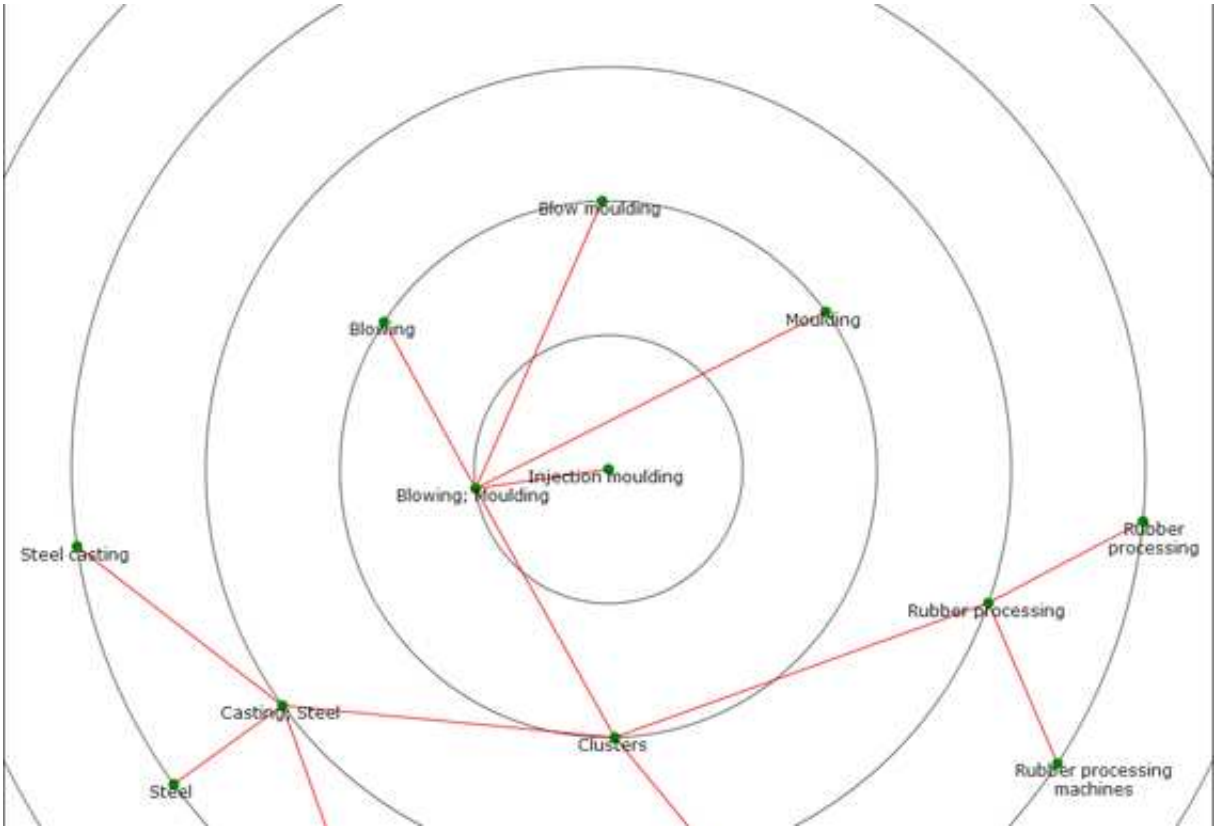

Figura 4.15: RELEVANT & GUI 2.

```
//create json variable from recovered clusters
for(int i=0; i<clusAfterXML.length; i++){
json += "   {\"id\":\""+clusAfterXML[i].getClusterGloss()+"\",\n" ;
json += "    \"name\":\"" +clusAfterXML[i].getClusterGloss()+"\",\n";
json += "    \"children\":[\n";

// save the keys of each cluster
 String key = clusAfterXML[i].getKeys().toString();
 while (key.length()>1)
 {
     //recover the first index from the first key
     int bb = key.indexOf("[");
     //System.out.println("indice di [ ="+bb);
     int cc = key.indexOf("]");
     //System.out.println("indice di ]= "+cc);
     String stringa = key.substring(bb+1, cc);
     //System.out.println(stringa);
     //separate by commas keys and save the values in a vector
     Vector vector_key = new Vector();
```

Figura 4.16: Estrazione Chiavi prima parte.

```
Vector vector_key = new Vector();
 while (stringa.length()>1)
 {
     //new string
     int dd = stringa.indexOf("");
     //System.out.println("indice di _ ="+dd);
     //end new string
     int ee = stringa.indexOf(",");
     //System.out.println("indice di , ="+ee);
     if (ee != -1){
         //if it isn't the end of the string
         String tmp = stringa.substring(dd, ee);
         //System.out.println(tmp);
         stringa = stringa.substring(ee+2);
         vector_key.add(tmp);}
     else{
         //System.out.println(stringa);
         vector_key.add(stringa);
         break;}

 }
```

Figura 4.17: Estrazione Chiavi seconda parte.

Conclusioni e Sviluppi futuri

Durante la stesura dell'elaborato si sono toccati diversi aspetti del problema dell'information overload.
Le problematiche generali che esso crea sono in prmis la perdita di tempo nel ricercare nozioni sintetiche e precise che soddisfino la sete di informazione dell' internauta, in secondo luogo il sovraccarico cognitivo che impedisce ad un utente di filtrare e selezionare le informazioni effettivamente rilevanti.
Si sono analizzati i primi tentativi per far fronte al fenomeno dell'information overload facendo riferimento più precisamente al news overload, passando dagli aggegatori di news, al Web Semantico e ai più avanzati sistemi di clustering di news. Le tecniche e gli algoritmi di clustering sfruttate dai diversi sistemi presentati dalla letteratura, sono stati oggetto di studio ed analisi di questo elaborato, e si è rivolta particolare attenzione al comportamente e alle preferenze di un utente per ogni sistema preso in considerazione.
Fra i vari sistemi, un intero capitolo è stato dedicato alla descrizione e allo studio di RELEVANT^{News}, che per certi versi concentra in sè gli aspetti migliori che si sono intravisti nei sistemi di clustering visti precedentemente.
RELEVANT^{News} offre all'utente la possibilità di scegliere i diversi parametri secondo i quali poter effettuare il clustering di news (un utente può scegliere le soglie dei vari algoritmi di calcolo di similarità presentati) e, con il presente lavoro di tesi, può avere una visualizzazione degli stessi. L'interfaccia grafica e l'animazione implementata permettono di avere una visione generale della disposizione spaziale dei cluster e delle news

contenute in esso, e danno la possibilità, una volta scelta la news di riferimento, di leggerla al centro del layout radiale e di presentare gli altri cluster attorno ad essa su cerchi concentrici rispettando la distanza fra il cluster di appartenenza e gli altri. L'utente può visualizzare i cluster creati immediatamente con RELEVANT, oppure decidere di rivedere una determinata organizzazione di cluster precedentemente salvata in un file XML.

Oltre all'aspetto pratico del mio elaborato l'analisi dei sistemi di clustering proposti in letteratura ha permesso di giungere ad alcune considerazioni relativamente importanti per uno sviluppo futuro di RELEVANT^{News}. Il sistema implementato dall' Università di Modena e Reggio nell'Emilia, a tutt'oggi non possiede algoritmi di clustering che seguano:

- *il tempo*: clusterizzare news che condividano la stessa data di pubblicazione sui giornali on line
- *il luogo*: clusterizzare news in base al luogo
- *fare uso dell'ipernimia e fissare una soglia*: quest'ultimo punto, forse quello più rilevante, potrebbe permettere al sistema di effettuare il calcolo delle matrici di similarità in tempi più brevi rispetto a quelli attualmente previsti; migliorerebbe in genere le prestazioni del sistema e si potrebbe pensare di memorizzare le matrici. Tale procedura permetterebbe di eseguire un calcolo accurato della "distanza"fra i vari cluster, e si potrebbe pensare di renderla visibili modificando le varie distanza fra i cerchi concentrici presenti nel layout che al momento risultano essere equidistanti gli uni dalgi altri.

Bibliografia

[1] Dalla filosofia all'informatica: Ontologia una concettualizzazione di un dominio condiviso.

[2] Wordnet an electronic lexical database. Cambridge, MA ; London, May 1998. The MIT Press.

[3] J. Allan, R. Gupta, and V. Khandelwal. Temporal summaries of new topics. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 10–18, New York, NY, USA, 2001. ACM Press.

[4] S. Banerjee, K. Ramanathan, and A. Gupta. Clustering short texts using wikipedia. In *SIRIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 787–788, New York, NY, USA, 2007. ACM Press.

[5] E. Banos, I. Katakis, N. Bassiliades, G. Tsoumakas, and I. Vlahavas. Personews: A personalized news reader enhanced by machine learning and semantic filtering. pages 975–982, 2006.

[6] S. Bergamaschi and F. Guerra. Overview of the theme topic, describing why it is important, timely, and relevant to ic.

[7] S. Bergamaschi, F. Guerra, M. Orsini, C. Sartori, and M. Vincini. Relevantnews: a semantic news feed aggregator. In G. Semeraro, E. D. Sciascio, C. Morbidoni, and H. Stoermer, editors, *SWAP*, volume 314 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[8] S. Bergamaschi, M. Orsini, F. Guerra, and C. Sartori. An automatic metadata extraction tool for knowledge management.

[9] S. Bergamaschi, C. Sartori, F. Guerra, and M. Orsini. Extracting relevant attribute values for improved search. volume 11, pages 26–35, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[10] I. Cantador, A. Bellogín, and P. Castells. News@hand: A semantic web approach to recommending news. pages 279–283, 2008.

[11] G. Cleuziou, L. Martin, and C. Vrain. Poboc: An overlapping clustering algorithm, application to rule-based classification and textual data. In *ECAI*, pages 440–444, 2004.

[12] A. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization: Scalable online collaborative filtering. In *16th International World Wide Web Conference*, May 2007.

[13] B. S. Everitt. Cluster analysis. A Hodder Arnold Publication, March 1993.

[14] S. I. Fabrikant, M. Ruocco, R. Middleton, D. R. Montello, and C. Jörgensen. The first law of cognitive geography: Distance and similarity in semantic space. 2002.

[15] A. F. Farhoomand and D. H. Drury. Managerial information overload. pages 127 – 131, October 2002.

[16] B. Fiorentini. Document clustering e nuovi motori di ricerca. pages 26–35, 2005.

[17] E. Gabrilovich, S. Dumais, and E. Horvitz. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 482–490, New York, NY, USA, 2004. ACM Press.

[18] A. Gulli. The anatomy of a news search engine. In *WWW 2005*.

[19] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. volume 17, pages 107–145, 2001.

[20] M. Hamdi. Information overload and customization. pages 9 – 12, Sept.-Oct. 2006.

[21] R. Hemayati, W. Meng, and C. T. Yu. Semantic-based grouping of search engine results using wordnet. In *APWeb/WAIM*, pages 678–686, 2007.

[22] http://flare.prefuse.org/apps/dependency_graph.

[23] http://it.wikipedia.org/wiki/Clustering.

[24] http://it.wikipedia.org/wiki/JSON.

[25] http://it.wikipedia.org/wiki/Polisemia.

[26] http://it.wikipedia.org/wiki/XML.

[27] http://www.msnbc.msn.com/id/24207533.

[28] A. Java, T. Finin, and S. Nirenburg. Semnews: A semantic news framework. In *AAAI*, 2006.

[29] C. D. Manning and H. Schütze. Foundations of statistical natural language processing. Cambridge, MA, USA, 1999. MIT Press.

[30] D. Radev, J. Otterbacher, A. Winkel, and S. Blair-Goldensohn. Newsinessence: summarizing online news topics. volume 48, pages 95–98, New York, NY, USA, October 2005. ACM Press.

[31] D. R. Radev, S. Blair-goldensohn, Z. Zhang, and R. S. Raghavan. Newsinessence: A system for domain-independent, real-time news clustering and multi-document summarization.

[32] P. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. volume 20, pages 53–65, Amsterdam, The Netherlands, The Netherlands, November 1987. Elsevier Science Publishers B. V.

[33] F. Schilder and C. Habel. From temporal expressions to temporal information: Semantic tagging of news messages. In *Proceedings of ACL'01 workshop on temporal and spatial information processing*, pages 65–72, Toulouse, France, 2001.

[34] A. Singh, S. Mukherjee, I.V.Ramakrishnan, G. Yang, and Z. Shah. Sean: A system for semantic annotation of web documents. 2003.

[35] A. Steinberg. Cs229 final project: Clustering news feeds with flock. December 16, 2005.

[36] A. Stepinski and V. O. Mittal. A fact/opinion classifier for news articles. In *SIGIR*, pages 807–808, 2007.

[37] K. T. Limitations in information processing in the human brain: neuroimaging of dual-task performance and working memory tasks. In *Progress in Brain Research*, pages 126: 95–102, 2000.

[38] S. Vadrevu, S. Nagarajan, F. Gelgi, and H. Davulcu. Automated metadata and instance extraction from news web sites. volume 0, pages 38–41, Los Alamitos, CA, USA, 2005. IEEE Computer Society.

[39] K. P. Yee, D. Fisher, R. Dhamija, and M. A. Hearst. Animated exploration of dynamic graphs with radial layout. In *INFOVIS*, pages 43–50, 2001.

Ringraziamenti

Desidero ringraziare la mia relatrice, la Prof. Sonia Bergamaschi, e gli Ingegneri Francesco Guerra e Mirko Orsini per la professionalità e la disponibilità mostratami durante i mesi di tirocinio.

Un grazie particolare a tutte le persone che ho avuto occasione di conoscere durante l'attività progettuale: Clara, Simone, Roberto e i ragazzi dell'OPTOLAB, che hanno reso meno pesante questi ultimi mesi di studio intenso. Nei ringraziamente non possono mancare gli amici di vecchissima data, quelli che insieme a me hanno iniziato il percorso di studi in ingnegneria.

In pole position **Maurizio** (che almeno in questa occasione mi riserverò dal chiamarlo Morris...) che dall'inizio alla fine del mio percorso di studi mi è stato sempre accanto, mi ha sostenuto in tutti modi possibili, spesso e volentieri lasciando da parte i suoi impegni per venire incontro alle mie difficoltà, le mille lacrime e i mille sorrisi, i McDonald's alle dieci di sera difficilissimi da digerire :) e i mille "dile, devi solo avere fiducia in te stessa perchè hai tutte le carte in regola per farcela"mi hanno aiutato ad andare avanti e a raggiungere questo obiettivo, sei e sarai sempre il mio angelo custode, grazie Zio!!

Da non dimenticare er Cilla e Giovanni, altri due mitici materani che conosco da una vita ormai, sempre pronti per una parola di conforto o per una battuta per tirarmi su di morale... grandi!

E i miei piccoli polentoni? Dove li lasciamo? Giorgia, Mary, Sara, Taty, Ricky (detto anche Zomb), Dany, Micky (meglio conosciuto come polacchese), quanto mi avete sop-

portato? Quanti consigli, quante consulenze, quante mani protese per farmi arrivare fino a qui... infinita è la mi gratitudine nei vostri confronti, e non credo esistano parole degne per descrivere le vostre splendide qualità!!
Nei ringraziamenti non può assolutamente mancare il mitico Zap, il brindisino dal cuore d'oro e dalla testa dura, se sono qui è anche per merito suo.
Per ultimi, ma non per questo meno importanti, vorrei ringraziare Matteo, che con la sua solarità e la sua simpatia ha fatto riemergere aspetti di me che per tanto tempo erano rimasti sopiti, e la mia famiglia.
Papà, Mamma, Plinio e Sheila, genitori e fratelli splendidi, modelli e punti di riferimento, custodi di valori preziosi e inestimabili, ho imparato dai vostri insegnamenti, dai vostri errori, mi avete aperto la strada e guidato tenendomi per mano; mi avete reso la Donna che sono oggi.

Io ho raggiunto un piccolo obiettivo, ma la mia vera fortuna è quella di avere voi al mio fianco, che siete i veri protagonisti di questa storia.

Grazie di vero cuore.