DIGITAL FUTURE
OR2 - Piattaforma Digital Future: Metodi e Tecniche

Metodi e tecniche per il Mediator
TR2.2

AUTORE: EXPRIVIA S.P.A

UNIONE EUROPEA
FONDO EUROPEO DI SVILUPPO REGIONALE

Ministero Sviluppo Economico

REGIONE PUGLIA
AREA POLITICHE PER LO SVILUPPO ECONOMICO
IL LAVORO E L'INNOVAZIONE

Regolamento regionale della Puglia per gli aiuti in esenzione n. 17 del 30/09/2014

Titolo II – Capo 1 "Aiuti ai programmi di investimento delle Grandi Imprese"

POR PUGLIA FESR 2014 – 2020

CONTRATTO DI PROGRAMMA

"Digital Future"

CODICE PROGETTO: CP12PA6

Diritti di autore e riservatezza
Questo documento è proprietà esclusiva della società Exprivia S.p.A. e non può essere riprodotto, anche in forma parziale, senza un'autorizzazione scritta della società stessa.

Indice dei contenuti

DIGITAL FUTURE **2**
INDICE DEI CONTENUTI **4**
INDICE DELLE FIGURE **5**
INDICE DELLE TABELLE **6**
1. INTRODUZIONE **7**
1.1. SCOPO E CAMPO DI APPLICAZIONE 7
1.2. ORGANIZZAZIONE DEL DOCUMENTO 7
2. L'AMBITO APPLICATIVO **9**
2.1. DESCRIZIONE GENERALE 9
2.2. INFORMATION RETRIEVAL 9
2.2.1. Definizione e caratteristiche *9*
2.2.2. Indicizzazione *12*
2.2.3. Indice Invertito *18*
2.2.4. Modelli di Information Retrieval *20*
2.2.5. Misure di Valutazione di Information Retrieval *27*
2.3. QUERY EXPANSION E RELEVANCE FEEDBACK 31
2.3.1. Applicazioni di NLP in IR *33*
2.4. WEB MINING E IR 40
2.4.1. Tecniche per la raccolta di informazioni dal Web *41*
2.5. DESCRIZIONE TECNICHE SPECIFICHE 41
3. IL MEDIATOR **44**
3.1. DESCRIZIONE GENERALE DELLA COMPONENTE 44
3.1.1. Le funzionalità *44*
3.2. ARCHITETTURA GENERALE 47
3.2.1. Descrizione delle componenti *49*
3.2.1. Integrazioni tra le componenti *57*
4. BIBLIOGRAFIA **58**

Indice delle figure

Figura 1 - Architettura IR....10
Figura 2 - Diagramma di contesto....44
Figura 3 - Architettura di riferimento....47
Figura 4 - Architettura logica del Mediator....48
Figura 5 - Esempio connettore RSS....51
Figura 6 - Esempio connettore Google Dork....53
Figura 7 - Esempio connettore Twitter Search....54
Figura 8 - Esempio output del composer, diagramma Entity Relationship....56
Figura 9 - Component Diagram....57

Indice delle tabelle

Tabella 1: Matrice termini-documenti 18
Tabella 2: Indice inverso 19

1. Introduzione

1.1. Scopo e campo di applicazione

Nel presente documento Exprivia si propone di descrivere in dettaglio una soluzione che nel seguito sarà denominata "Mediator", sviluppata come macro-componente della piattaforma di Digital Future, finalizzata a supportare gli esperti di dominio nei vari processi decisionali atti alla ricerca e al consolidamento di soluzioni innovative finalizzate al miglioramento del proprio business.

In linea con quanto espresso, finalità del Mediator è pertanto quella di consentire l'acquisizione di informazioni provenienti dal web in modo più rapido ed efficace, in termini di inerenza con i risultati attesi dall'utente, rispetto ad un approccio tradizionale basato su attività manuali di ricerca.

Per ottemperare a quanto fin qui espresso il Mediator dovrà mettere a disposizioni servizi di:

- supporto allo sviluppo di iniziative di business e di approntamento tecnologico;
- analisi e valutazione di brevetti e pubblicazioni scientifiche per la ricerca di soluzioni tecnologiche (technology scouting)

Prima di addentrarci nei dettagli più prettamente tecnici ed implementativi del Mediator, al fine di comprenderne le specifiche peculiarità, proponiamo nel seguito:

1. un'analisi dettagliata del dominio in cui tale componente si colloca e dei relativi campi di applicazioni possibili;
2. un'analisi dello stato dell'arte volta alla individuazione di metodologie e tecniche già presenti sul mercato per la realizzazione del componente software nonché una gap analysis volta altresì alla individuazione di eventuali carenze tecnologiche da colmare per realizzare il componente Mediator;
3. una descrizione dei servizi previsti all'interno del Mediator;
4. l'architettura logica del componente insieme alle tecnologie proposte per l'implementazione della stessa;
5. possibili scenari e sviluppi futuri.

1.2. Organizzazione del documento

Il documento è strutturato come segue:

- *Capitolo 1:* Introduzione
- *Capitolo 2:* Analisi del dominio
- *Capitolo 3:* Mediator

- *Capitolo 4:* Architettura logica del Mediator
- *Bibliografia*

2. L'ambito applicativo

2.1. Descrizione generale

Come introdotto nella sezione precedente, l'elemento peculiare del Mediator è quello di fornire servizi che permettano di aiutare i processi di business e di scouting tecnologico per una azienda o un candidato imprenditore per iniziare nuove attività imprenditoriali o per estenderne di già presenti.

Un aspetto importante da tenere presente, è l'accessibilità ad informazioni che possono essere rilevanti per il Mediator. L'unica fonte aperta, quindi liberamente accessibile, è il Web, dove i sistemi di indicizzazione dei documenti, quali ad esempio Google e Yahoo, hanno modificato in maniera determinante la modalità con cui si accede alle informazioni di interesse. I motori di ricerca sopra menzionati, così come altri disponibili, utilizzando sistemi di Information Retrieval utilizzati per raccogliere informazioni e per consentire un accesso molto rapido alle informazioni. Inoltre grazie alla possibilità di essere personalizzati, permettono la raccolta di informazioni e di classificarle sulla base di domini specifici, in base agli interessi degli utenti. Spesso però le attività di reperimento delle informazioni relative ad uno specifico dominio possono comportare delle difficoltà; in questo caso si rende necessario mettere in piedi dei sistemi web intelligenti basati sulla semantica, che di fatto siano in grado di fornire informazioni in linea con gli stessi interessi degli utenti.

In tale capitolo verranno approfondite le principali tecniche applicate per ottenere un sistema di Information Retrieval e, alla fine dello stesso la relazione rispetto al Mediator.

2.2. Information Retrieval

2.2.1. *Definizione e caratteristiche*

In ambito accademico l'Information Retrieval (IR)1 viene definito come la ricerca di dati non strutturati, generalmente documenti, o di informazioni specifiche rivenienti da un fabbisogno informativo presentato da un utente ed espresso attraverso query (interrogazioni) definite dall'utente.

Il dominio dell'Information Retrieval (IR)[2] può essere visto, in una certa misura, come un'applicazione di successo del Natural Language Processing (NLP). La velocità e la scalabilità nel reperimento di informazioni

[1] Bing Liu - Web Data Mining Exploring Hyperlinks, Contents, and Usage Data - ISBN 978-3-642-19459-7

[2] Manning, C. D., Raghavan, P., Sch utze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (UK), 2008.

dal Web in tutto il mondo è stato reso possibile grazie dalla ricerca libera e ai motori di ricerca presenti in commercio. Questi strumenti sono utilizzati da circa l'85% dei navigatori Web nella ricerca di informazioni specifiche.[3]

Tecnicamente l'Information Retrieval (IR) studia l'acquisizione, l'organizzazione, la memorizzazione, il reperimento e la distribuzione di informazioni quali documenti, pagine web, cataloghi on line e oggetti multimediali[4]. Nella figura che segue viene proposta una tipica architettura IR:

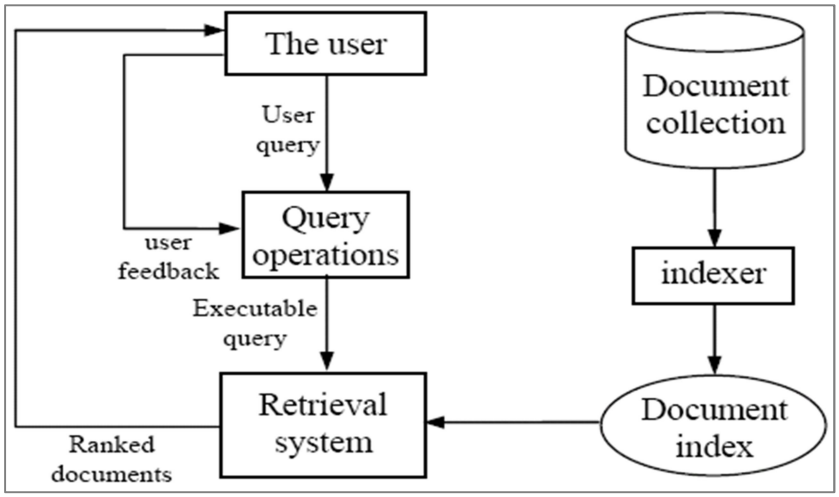


Figura 1 - Architettura IR

Una delle prime e più importanti caratteristiche dell'IR è rappresentata dal fatto che un sistema IR deve poter gestire descrizioni imprecise e incomplete relative sia alle necessità dell'utente sia ai documenti richiesti, giungendo nella maggior parte dei casi, ad una impossibilità nel calcolare una precisa e non ambigua rappresentazione della pertinenza dei documenti rispetto ad una data query. Spesso, infatti, l'utente tende ad inserire delle query nei motori di ricerca o in modo ambiguo e poco descrittivo delle sue reali necessità o in una definizione troppo generale. In pratica gli utenti non descrivono le loro esigenze di informazione con tutti i dettagli necessari, preferendo che sia il sistema a suggerire loro alcune risposte, selezionando successivamente quelle maggiormente attinenti alle loro necessità.

Il processo di ricerca, pertanto, dovrebbe essere visto più come un approccio per la risoluzione dei problemi del tipo "trial and error" (prova e sbaglia) piuttosto che come un approccio diretto "query-response" (domanda-risposta). La corrispondenza tra la query e l'elemento informativo non è deterministica: il sistema fornisce le migliori risposte possibili (migliori corrispondenze) stimando le loro probabilità di

[3] Wolfram, D., Spink, A., Jansen, B.J., Saracevic, T.: Vox populi: The public searching of the web. Journal of the American Society for Information Science and Technology, 52 (2001) 1073{1074.

[4] Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley, Reading (MA), 1999.

pertinenza rispetto alla query. La difficoltà principale nel recuperare i documenti rilevanti per una query risiede in tre principali aspetti rivenienti fondamentalmente da alcune proprietà di tutte le lingue naturali, vale a dire ***polisemia***, ***sinonimia*** e, in misura minore, rivenienti dal modo di uso di regole che servono a scrivere bene, quali ***errori di ortografia e variazioni.***

In tutte le lingue naturali, una data parola può avere più di un significato preciso (polisemia). Alcuni di questi sensi potrebbe essere correlati, ma in altri casi, il significato sottostante potrebbe variare notevolmente. Ad esempio con la parola "Java", i significati possibili sono ben poco correlati tra loro (un'isola, una compagnia, una danza, un pollo domestico, un linguaggio di programmazione per computer). Se questa parola viene inserita in una query, il sistema di ricerca tenderà a fornire risposte errate.

Mentre la polisemia potrebbe essere posta in analogia al modo di far riferimento contemporaneamente a facce diverse di una stessa moneta, se invece si dovesse invece far riferimento in maniera differente alla stessa faccia della stessa moneta si incorre alla sinonimia. Di fatto, quindi, la sinonimia definisce la possibilità che diverse parole o espressioni possono essere utilizzate per fare riferimento allo stesso oggetto. Ad esempio, per fare riferimento a un determinato incidente d'auto, possiamo usare i seguenti termini: "incidente", "evento", "situazione sfortunata", etc.

Come risultato di questi due problemi principali, come anche dimostrato empiricamente in[5], la probabilità che due persone utilizzino lo stesso termine per descrivere lo stesso oggetto è sotto 0.20. Pertanto la domanda che sorge spontanea è: "come può un sistema di ricerca, in un ambiente in cui è presente tale variabilità, funzionare correttamente e con livelli di prestazione ragionevoli?"

Tutti i sistemi di ricerca presumono che parole simili tendano a corrispondere a simili significati. Quindi quando una query condivide molti termini con un documento, questo documento è visto come una risposta appropriata alla query corrispondente. I sistemi di ricerca sono quindi basati su un approccio semplice: estrarre parole dai documenti e dalle query, confrontare i due gruppi di parole e classificare il documento secondo questo confronto, potenzialmente utilizzando elementi aggiuntivi, come ad esempio il "PageRank". Come mostrato da Hawking & Robertson[6] con l'aumentare delle dimensioni del corpus dei documenti, le prestazioni raggiunte da tale processo tendono ad aumentare.

Nel seguito proponiamo una descrizione più dettagliata di un classico sistema di IR focalizzandoci sui seguenti argomenti:

---

[5] Furnas, G., Landauer, T.K., Gomez, L.M., Dumais, S.T. : The vocabulary problem in human-system communication. Communications of the ACM, 30 (1987) 964{971.

[6] Hawking, D., Robertson, S.: On collection size and retrieval effectiveness. IR Journal, 6 (2003) 99-150.

- I principali aspetti di un processo di indicizzazione utilizzato per costruire la rappresentazione delle query e dei documenti.
- I principali modelli di IR.
- Le metodologie di valutazione di un sistema di IR.

Le query espresse dall'utente possono essere di vario tipo:

- **Keyword Query**, interrogazioni espresse mediante parole chiave (es.: "A320" se si desidera reperire documenti relativi ad un particolare modello di aereo dell'Airbus).
- **Boolean Query**, interrogazioni contenente operatori booleani quali AND, OR, NOT (es.: "stanford university" AND "university palo" AND "palo alto").
- **Phrase Query,** interrogazioni basate su sequenze di termini (es.: "President of the United States").
- **Proximity Query**, interrogazioni in cui è richiesto di ricercare documenti in cui un termine si trova nelle vicinanze di un altro (es.: "Gates NEAR Microsoft").
- **Full document Query**, interrogazioni che vengono eseguite a fronte di una collezione di documenti memorizzati su un database di tipo full text[7]. In questo caso la ricerca della documentazione di interesse viene eseguita esaminando tutte le parole contenute in ogni documento cercando di trovare una corrispondenza con le parole inserite dall'utente all'interno della sua query.
- **Natural Language Query**, interrogazioni espresse in linguaggio naturale ossia senza l'utilizzo di una sintassi o di un formato particolare (es.: "What is the most important cathedral of Paris?").

2.2.2. *Indicizzazione*

Questo approccio è stato sviluppato dall'inizio degli anni '60[8] e consente di elaborare le enormi quantità di informazioni disponibili online permettendo una "catalogazione" dei contenuti scansionabili.

E' importante tener presente che la maggior parte dei sistemi IR esistenti si basa sull'indicizzazione automatica dei documenti e delle query. Prima che i documenti di una collezione possano essere utilizzati per il reperimento, viene eseguito un processo preliminare sui testi. Sui documenti di testo tradizionali (senza tag HTML), normalmente vengono eseguiti vari interventi quali, ad esempio, rimozione di *stop word*, manipolazione dei numeri, i segni di punteggiatura ecc. Tuttavia le pagine web sono differenti dai documenti di testo tradizionali e per esse sono pertanto previste delle operazioni di pre-processing ad hoc, quali:

---

[7] https://en.wikipedia.org/wiki/Full-text_database

[8] Salton, G. (Ed): The SMART Retrieval System. Experiments in Automatic Document Processing. Prentice-Hall, Englewood Cliffs (NJ), 1971.

- *identificazione dei vari campi di testo*: In HTML vi sono vari campi di testo, come per esempio titolo, metadati e corpo. La loro identificazione permette al sistema di reperimento a trattare i termini dei vari campi in modo differente. Per esempio nei motori di ricerca i termini che appaiono nel campo del titolo di una pagina vengono considerati più importanti dei termini che appaiono in altri campi e vengono assegnati pesi maggiori perché il titolo è solitamente una descrizione concisa della pagina. Nel testo del corpo ai termini enfatizzati vengono dati un peso maggiore (i tags <h1>, <h2>,..., <b>, ecc.).
- *identificazione di testi "ancorati"*: questi testi, associati all'iperlink, vengono trattati con una particolare importanza nei motori di ricerca perché il testo ancorato spesso rappresenta una descrizione più accurata di un'informazione contenuta nella pagina a cui suo link punta. Il caso in cui l'iperlink punti ad una pagina esterna, è una situazione particolarmente importante in quanto fornisce una descrizione di riepilogo riguardo alla pagina data da persone diverse da autore/proprietario della pagina corrente.
- *rimozione dei tag HTML*: essa può essere affrontata mediante il calcolo del punteggio di similarità. L'HTML è una presentazione visuale del linguaggio e, nella tipica pagina HTML, l'informazione è presentata in blocchi rettangolari. La semplice rimozione dei tags HTML potrebbe causare problemi comprensione del testo e conseguenti problematiche nel determinare la rispondenza di una determinata pagina web rispetto ad una query immessa dall'utente.
- *identificazione dei blocchi di contenuti principali*: Una tipica pagina web contiene una grande quantità di informazioni che non rappresenta contenuto principale della pagina. E' stato dimostrato che i risultati di ricerca e di data mining possono essere migliorati notevolmente se vengono utilizzati solamente i blocchi principali nei contenuti.

Nei paragrafi seguenti, si descrivono i quattro passaggi fondamentali eseguiti da un algoritmo di indicizzazione automatico nella fase di elaborazione di documenti:

- Analisi della struttura e tokenizzazione.
- Rimozione delle stopwords.
- Normalizzazione morfologica.
- Pesatura.

2.2.2.1. *Analisi della struttura e Tokenizzazione*

In questo primo passaggio, i documenti vengono analizzati in modo da riconoscerne la struttura (titolo, abstract, sezione, paragrafi). Per ogni struttura logica rilevante, il sistema opera nella "tokenizzazione " del testo, cioè segmenta le frasi in token di parole. Questa procedura, che sembra relativamente semplice, è

spesso resa complicata dall'utilizzo delle abbreviazioni in un testo e può portare il sistema ad una erronea rilevazione del confine di una determinata frase; inoltre si rende necessario analizzare più approfonditamente i casi in cui sono presenti numeri, caratteri speciali o lettere maiuscole. Ad esempio: nelle espressioni inglesi "don't ", "I'd", "John's" abbiamo uno, due o tre token? oppure, nell'espressione "afro-americano", includiamo il trattino nella tokenizzazione o consideriamo solo uno o due token?

Per i numeri, in realtà, non esiste una regola ben definita e applicabile in ogni contesto: possiamo semplicemente ignorarli o includerli come unità indicizzata. Un'alternativa è indicizzare tali entità in base al loro tipo, ad esempio utilizzando i tag "date", "currency", ecc. al posto di una determinata data o quantità di denaro. Relativamente le lettere maiuscole, esse devono essere trasformate in minuscole; quindi, il titolo "Esportazione di automobili dalla Francia" è tradotto nella sequenza di parole "esportazione", "di", "automobili", "dalla" e "francia".

Quando il gruppo di token che costituisce una frase è sostenuto da una struttura grammaticale, viene implementata la cosiddetta etichettatura (***tagging***) delle parti del discorso (**Part-of- Speech (POS)**) mediante cui ogni parola è etichettata da una particolare categoria lessicale che determina il ruolo svolto dalla specifica parola in un determinato frammento di testo.

2.2.2.2. *Rimozione delle StopWords*

In una seconda fase, parole molto frequenti come gli articoli determinativi "il", le preposizioni come "da", le congiunzioni come "e", i pronomi come "tu" e alcune forme verbali come "è" ecc.., compaiono in elenchi di *stopword*. Le stopword, anche denominate parole vuote appunto ad indicare che non hanno alcun significato, rappresentano il rumore nel processo di recupero e effettivamente danneggiano le prestazioni di un sistema di Information Retrieval, poiché non consentono di discriminare tra documenti rilevanti e non rilevanti. Si procede pertanto in una rimozione delle stesse, la cui rimozione appunto è motivata essenzialmente dalle seguenti considerazioni:

- La prima fondamentale considerazione si basa sulla necessità di reperire solo documenti che rispondono ad una query esclusivamente sulla base di parole contenenti un contenuto reale e significativo: recuperare un documento solo perché contiene le parole della query "essere", "in" e "il" non rappresenta infatti una strategia di ricerca intelligente.
- La seconda importante considerazione è rappresentata dal fatto che la rimozione delle Stopword consente di ridurre la dimensione del corpus di documenti indicizzato, anche di percentuali elevate comprese tra il 30% e il 50%.

Sebbene gli obiettivi sembrino chiari, non esiste una metodologia chiara e completa per determinare un elenco di stopword valido a priori[9]. Ad esempio, il sistema SMART [10] ha 571 parole nella sua lista di stopwords, mentre i servizi di informazione DIALOG propongono l'utilizzo solo di nove termini (ovvero "an", "e", "by", "for", "from", "of", "the ","to", e "with"). Inoltre, alcune espressioni, come "The Who", "and-or gates", "vitamin A", formate da parole che di solito si trovano nell'elenco delle stopwords, sono molto utili per specificare più precisamente ciò che l'utente desidera.

Allo stesso modo, dopo aver convertito tutti i caratteri in lettere minuscole, alcune ambiguità possono essere introdotte come, ad esempio, trasformando l'espressione "US Citizen" in "us citizen" la parola "us" sarebbe interpretata come pronome e pertanto eliminata (in quanto intesa come appartenente alla lista delle stop words) determinando una erronea elaborazione del testo. Lo stesso avverrebbe trasformando l'espressione "IT Scientist" in "it scientist" in cui "it" verrebbe interpretato anche in questo caso come pronome. La strategia per quanto riguarda il trattamento delle stopwords in questo caso è quella di identificare "US" e "IT" non come pronomi ma come nomi propri attraverso un ulteriore processo denominato *part-of-speech tagging* già menzionato nel precedente paragrafo. Per le motivazioni sopra espresse la maggior parte dei motori di ricerca presenti in commercio utilizza un numero ridotto di stopwords.

2.2.2.3. *Normalizzazione Morfologica*

Come terzo passaggio, la procedura di indicizzazione utilizza il processo di normalizzazione morfologica nel tentativo di far confluire varianti di parole nella stessa radice.

Le procedure di **stemming**, che mirano a identificare la radice di una parola e ad usarla al posto della parola stessa, sono di gran lunga la più comune procedura di normalizzazione morfologica utilizzata in IR. In dettaglio lo *Stemming* può essere definito come l'identificazione della radice di una parola: infatti "stem" vuol dire stelo, e in questo senso lo stem di una parola è lo stelo su cui si applicano le varianti che danno luogo alle derivazioni linguistiche della parola. Il raggruppamento delle parole con la stessa radice sotto lo stesso stem (o termine di indicizzazione) possono incrementare di molto il tasso di successo nel ricercare le corrispondenze tra i documenti e una data query. Una tale procedura automatica può quindi essere uno strumento prezioso nel migliorare l'efficacia nel reperimento delle informazioni di interesse, assumendo

[9] Fox, C.: A stop list for general text. ACM - SIGIR Forum, 24 (1989) 19-21.

[10] Salton, G. (Ed): The SMART Retrieval System. Experiments in Automatic Document Processing. Prentice-Hall, Englewood Cliffs (NJ), 1971.

che le parole con la stessa radice si riferiscano alla stessa idea o concetto, e possono quindi essere indicizzate nella stessa forma.

Pertanto un sistema di Information Retrieval deve poter riconoscere le varianti e ottenere lo stelo, ovverosia la radice. Mediante lo stemming, per ciascuna di queste parole viene costruita una stringa che ne rappresenta la radice linguistica. In questo modo le parole che provengono dalla stessa radice linguistica sono ricondotte allo stesso stem.

Lo stemming è pertanto estremamente utile nella fase di pre-processing in quanto:

- Migliora notevolmente l'efficacia di un sistema di Information Retrieval e Text Mining, favorendo la rilevazione di parole simile all'interno di un testo e migliorando di conseguenza la recall (termine il cui significato sarà più chiaro nei paragrafi che seguono).
- Riduce la dimensione del materiale da indicizzare: combinando tra loro parole con la stessa radice è infatti possibile ridurre la dimensione del materiale da indicizzare anche del 40/50%.

Le principali tecniche di stemming sono:

- ***Remove ending*** (rimozione della parte finale della parola). Ad esempio nella lingua inglese alcuni classici esempi di remove ending sono la rimozione della s finale, di es finale o di ing finale.
- ***Trasformazione delle parole:*** ad esempio nella lingua inglese se una parola termina con "ies" ma non con "eies" o "aies" allora ***ies*** può essere sostituito da ***y***.

2.2.2.4. *Pesatura*

Come descritto in precedenza, un sistema IR segmenta automaticamente una determinata frase in parole attraverso il processo di tokenizzazione , rimuovendo quelle più frequenti ed eliminando i suffissi al fine di ottenere delle unità di indicizzazione. Ad esempio, dalla frase "In 1969, the IBM-360 computer was one of the first third generation computers", possiamo ottenere le seguenti unità di indicizzazione: "IBM-360", "first", "third", "generat", "computer". Questo risultato corrisponde a uno schema di indicizzazione binaria all'interno del quale ciascun documento è rappresentato da un insieme di parole chiave (stemmed) senza alcun peso assegnato. Spesso però le restrizioni logiche binarie possono essere troppo restrittive per l'indicizzazione di un documento o di una query. Non è sempre chiaro se un documento debba essere o meno indicizzato da un determinato termine. In quest'ottica la *pesatura* dei termini crea una distinzione tra i termini e nel contempo incrementa la flessibilità dell'indicizzazione. Quindi si dovrà procedere assegnando peso più elevato alle features più "importanti" e peso inferiore a quelli marginali. Per pesare in modo appropriato ogni unità di indicizzazione, si possono considerare tre componenti: la term frequency

(frequenza del termine), la "document frequency" (la frequenza del documento) e la lunghezza del documento[11].

Innanzitutto, si può supporre che se un'unità di indicizzazione appare più spesso in un documento deve avere una maggiore importanza nel descrivere il contenuto semantico del documento stesso. E' possibile misurare questo calcolando la frequenza del termine (cioè il suo numero di occorrenze all'interno di un documento), definito con la notazione ***tf (term ferquency)***. Quindi, se un termine occorre tre volte in un documento, il suo tf sarà 3. Naturalmente, si può considerare altri semplici varianti, soprattutto se si parte dal presupposto che la presenza di un dato termine in un documento è un evento raro. Pertanto, potrebbe essere una buona norma dare più importanza alla prima occorrenza del termine piuttosto che alle altre calcolando il componente tf come ***log (tf + 1)*** o come 0,5 + 0, 5 * (tf /max (tf)]. In quest'ultimo caso, la procedura di normalizzazione è ottenuta dividendo tf per il massimo valore tf per qualsiasi termine in quel documento.

Come seconda componente di pesatura, si può considerare che se tali termini sono presenti molto spesso nel corpus di documenti non ci aiutano a discriminare tra documenti rilevanti e non rilevanti. Il concetto appena espresso è rappresentato dalla nozione di frequenza di un termine nel corpus (cioè il numero di documenti in cui appare un termine), rappresentato dalla notazione ***df*** (***document frequency***).

Più precisamente, utilizzeremo il logaritmo della frequenza inversa del documento (denotato da ***idf = log (n / df)***, con n che indica il numero di documenti presenti nel corpus, con conseguente maggiore peso per le parole rare e meno peso per più quelle frequenti[12]. Con questo componente, se un termine occorre in ogni documento (df = n), il suo peso sarà log (n/n) = 0, e quindi dobbiamo ignorarlo. Dall'altra parte, quando un termine appare in un solo documento (df = 1), il suo peso raggiungerà il valore massimo nel corpus, in quanto log (n / 1) = log (n).

Per integrare entrambi i componenti (tf e idf), possiamo moltiplicare il peso corrispondente all'importanza del termine di indicizzazione all'interno del documento (tf) per la sua rilevanza nell'intero corpus (idf). Otteniamo così la ben nota formula di ***tf- idf***.

Infine, di solito si considera che la presenza di un termine in un documento più corto fornisce elementi più importanti rispetto alla sua presenza in un documento più lungo. Questo fenomeno può essere tenuto in considerazione tenendo conto della lunghezza del documento nel calcolo della pesatura di un termine, che

[11] Salton, G., Buckley, C.: Term weighting approaches in automatic text retrieval. Information Processing & Management, 24 (1988) 513-523.

[12] Sparck Jones, K.: A statistical interpretation of term specicity and its application in retrieval. Journal of Documentation, 28 (1972) 11{21.

di solito viene ottenuto confrontando la lunghezza di un documento rispetto alla lunghezza media dei documenti nel corpus (indicato con ***avdl***).

2.2.3. *Indice Invertito*

In un sistema IR, qualsiasi richiesta di informazione viene elaborata facendo uso dell'indice. L'indice è una struttura informativa che raccoglie le posting list relative ad un insieme dato di descrittori.

L'indicizzazione è l'unico modo per rendere il reperimento efficiente e allo stesso tempo efficace. Se un documento non è indicizzato allora non può essere reperito. Gli strumenti utilizzati per l'indicizzazione hanno un impatto diretto sui modelli e metodi di reperimento, e di conseguenza anche sugli esiti della valutazione di un sistema IR.

I problemi dell'indicizzazione riguardanti le pagine web sono eterogenei per la loro vastità di contenuti, struttura, media e lingue.

I metodi di indicizzazione sono metodi di base per la rappresentazione del contenuto informativo dei documenti di una collezione e delle interrogazioni poste dagli utenti finali, sono le soluzioni principali per la realizzazione degli indici. Infine, spiega i meccanismi essenziali di un agente di ricerca per il web. Mediante i descrittori, un sistema IR rappresenta il contenuto informativo di ciascun documento della collezione. L'insieme dei descrittori è l'unico insieme a disposizione dell'IRS per descrivere il contenuto informativo.

Il principale metodo di indicizzazione utilizzato è rappresentato ***dall'indice invertito*** [13].

L'indice invertito (detto anche file invertito) di un insieme di documenti è fondamentalmente una struttura dati che associa ad ogni termine distinto contenuto nell'insieme di documenti un elenco di tutti i documenti che contengono quel termine.

Nella tabella sottostante viene mostrata un esempio molto semplice di mappatura tra 4 documenti e l'insieme dei termini in essi contenuti.

	Indexing Terms				
Document	linguistic	document	search	compute	retrieval
D_1	1	0	0	0	0
D_2	1	1	0	1	1
D_3	0	1	1	0	1
D_4	1	0	1	0	1

Tabella 1: Matrice termini-documenti

[13] Zobel, Moffat & Ramamohanarao 1998

Tuttavia al fine di rendere più veloce il reperimento dei documenti afferenti una data query, le informazioni non vengono memorizzate in una struttura matriciale come quella esposta in Tabella 1.

Infatti, man mano che la collezione cresce anche il numero dei termini di indicizzazione tende ad aumentare rapidamente. Pertanto, è prassi trovarsi di fronte anche a decine di milioni di termini per insiemi contenenti anche diversi milioni di documenti. In questo contesto, per verificare se i documenti rispettano il vincolo logico imposto dalla query, abbiamo bisogno di accedere rapidamente agli identificativi dei documenti indicizzati con le parole chiave ricercate nella query. Per ottenere questo, il sistema memorizza in un "file invertito" la mappatura tra ciascun termine presente nell'insieme di documenti e la lista di documenti in cui esso è contenuto (per semplicità i documenti vengono identificati con il prefisso D). Ad esempio, la Tabella 2 mostra l'indice invertito corrispondente alla Tabella 1. Varie tecniche sono state suggerite[14] al fine di ridurre i requisiti di memorizzazione, accelerare l'elaborazione o consentire l'uso di frasi nelle query oltre ai semplici operatori logici (ad esempio, "New York City ") e l'uso di operatori di prossimità (ad esempio attraverso l'utilizzo dell'operatore di adiacenza (es.: "New York" ADJ "city").

Indexing term	Sequence of document identifiers
linguistic	D_1, D_2, D_4
document	D_2, D_3
search	D_3, D_4
compute	D_2
retrieval	D_2, D_3, D_4

Tabella 2: Indice inverso

Attraverso la tecnica dell'indice invertito, è possibile impiegare un tempo costante per reperire i documenti che contengono un determinato termine di ricerca contenuto nella query e inoltre è più facile gestire query con più termini.

Utilizzando la tecnica dell'indice invertito, la ricerca di documenti afferenti una determinata query q avviene attraverso i seguenti passaggi:

- ***Ricerca del vocabolario***: si tratta di ricercare ogni termine/parola della query q nell'indice invertito.
- ***Aggregazione dei risultati***: mettere insieme i risultati per trovare i documenti che contengono tutte o alcune delle parole / termini presenti nella query q.

[14] Zobel, J., Moffat, A.: Inverted file for text search engines. ACM Computing Surveys,38 (2006) 1-56.

- ***Calcolo dello score di classificazione***: per classificare i documenti / pagine risultanti, possono essere utilizzati i seguenti approcci:
    - classificazione basata sul contenuto
    - classificazione basata su link

Sulle tecniche utilizzate in IR per calcolare lo score di classificazione rimandiamo al paragrafo che segue.

2.2.4. *Modelli di Information Retrieval*

Un modello IR è un insieme di costrutti di rappresentazione del contenuto informativo dei documenti e delle query e in tal senso stabilisce i criteri con cui è definita la pertinenza di un documento ad una determinata query espressa dall'utente.

I principali modelli IR sono:

- **I modelli booleani (Boolean Models)**, in cui l'informazione associata ad un documento può assumere solo due valori: rilevante o non rilevante. La valutazione di tale valore dipende dalla presenza o assenza di un termine richiesto. Possono essere utilizzati gli operatori boolean AND, OR e NOT per combinare più termini;
- **I modelli vettoriali (Vector-Space Models)**, i documenti vengono rappresentati in uno spazio ipotetico in cui le dimensioni sono le parole e i documenti sono vettori in tale spazio. La ricerca avviene per similarità tra una query rappresentata in questo spazio e i vettori dei documenti;
- **I modelli linguistici probabilistici (Probabilistic Models)**, in cui vengono valutate le probabilità di rilevanza avendo a disposizione i documenti e la query di ricerca.

Per rispondere alle esigenze di quanto prefissato progettualmente e a seguito di valutazioni specifiche rivenienti da diversi studi e test, si è pensato di utilizzare SOLR, il quale utilizza come modello una versione "boolean" modificata per fornire un peso ad ogni risposta e di conseguenza un ordinamento nelle risposte. Per offrire una maggiore visione, si ritiene comunque indispensabile fornire dettagli sui modelli sopra citati per poi meglio comprendere le scelte specifiche che saranno adottate.

2.2.4.1. *Modello booleano*

Nel ***modello booleano*** i termini della query sono combinati logicamente utilizzando gli operatori booleani AND, OR e NOT. Ad esempio: ((data AND mining) AND (not text)) .

Dato un query booleano, il sistema reperisce ogni documento che rende l'interrogazione logicamente vera. Pertanto il reperimento si basa sul criterio della decisione binaria. Intuitivamente si dice accoppiamento esatto ("***Exact Match***"). Uno degli inconvenienti principali dell'uso del modello logico è riconducibile al sovraccarico cognitivo necessario all'utente per eliminare la confusione tra gli operatori logici AND, OR e

NOT e le preposizioni "e", "o" e "non" della lingua naturale. Occorre pertanto che gli utenti che utilizzando tali modelli conoscano bene i criteri relativi alla logica booleana.

Nei modelli booleani i documenti e le query vengono trattati come un contenitore di parole ("***bag of words***") o di termini e non sono prese in considerazioni sequenze di parole.

Dato un insieme di documenti D, indichiamo con V = $\{t_1,t_2,t_3,.....t_V\}$ l'insieme di parole/termini distinti all'interno di un insieme di documenti. V è detto "vocabolario". Ad ogni termine ti di un documento Dj ∈ D viene associato un peso wij > 0. Se il termine ti non appare nel documento dj, il peso wij = 0. Ogni documento viene rappresentato con la seguente tupla:

$$d_j = (w_{1j}, w_{2j}, ..., w_{|V|j})$$

I risultati ottenuti utilizzando questo tipo di modello sono, purtroppo, piuttosto scadenti in quanto non viene presa in considerazione la frequenza con cui il termine occorre all'interno dei documenti.

2.2.4.2. *Vector Space Model*

Anche nel **modello vettoriale** i documenti sono trattati come un contenitore di parole ("bag of words") o termini, ma in questo caso ogni documento è rappresentato come un vettore pesato, i cui pesi dei termini sono compresi tra 0 e 1 e vengono calcolati in base ad una variazione dello schema di Frequenza di Termini (TF-IDF o TF).

Nello schema TF (***Term Frequency Scheme***), il peso wij del termine t_i del documento d_j è rappresentato dal numero volte che t_i compare nel documento dj, al quale è anche possibile applicare la normalizzazione. Lo svantaggio dello schema di TF e che non considera la situazione dove un termine può comparire in molti documenti della collezione, come fosse un termine poco discriminativo.

Se N è il numero totale di documenti nel sistema o nella collezione, e df_j il numero di documenti in cui il termine t_j compare almeno una volta. Sia f_{ij} la frequenza empirica del termine t_i nel documento d_j. Allora la frequenza di termini normalizzata tf_{ij} in d_j é data da:

$$tf_{ij} = \frac{f_{ij}}{\max\{f_{1j}, f_{2j}, ..., f_{|V|j}\}}$$

in cui il massimo è calcolato su tutti i termini che compaiono all'interno del documento dj.

L'inverso della frequenza è dato dalla formula:

$$idf_i = \log \frac{N}{df_i}$$

secondo la quale se un termine compare in molti documenti, è poco rilevante e quindi non discriminante.

La pesatura **TF-IDF** è invece data dalla seguente formula:

$$w_{ij} = tf_{ij} \times idf_i.$$

La pesatura TD-IDF è data pertanto dalla combinazione tra la frequenza del termine nel singolo documento e l'inverso della frequenza del termine in tutti i documenti del corpus. Vengono pertanto dapprima contate le occorrenze di un termine all'interno di un documento; in base a questo calcolo l'elevata frequenza di un termine all'interno di un documento indica che il documento tratta argomenti relativi a quel termine. A seguire vengono contati i documenti appartenenti all'insieme in esame che contengono quella parola: se una determinata parola compare tante volte nei documenti del corpus viene considerata poco discriminante quindi poco rilevante.

Una query q potrebbe essere rappresentata esattamente allo stesso modo come un documento in un insieme di documenti. Il peso dei termini w_{iq} di ciascun termine t_i in q può anche essere calcolato come in un documento normale o leggermente differente.

Nel modello vettoriale, a differenza del modello booleano, non viene utilizzato il criterio della decisione binaria ma i documenti sono classificati in base allo score di pertinenza rispetto alla query. Un sistema per determinare il grado di rilevanza è quello di calcolare l'indice di similarità della query q per ciascun documento dj della collezione di documenti D. A seconda del contesto, si potrebbe utilizzare qualunque tipo di indice di similarità per calcolare lo score di rilevanza. Una misura di similarità particolarmente adottata è quella basata sulla similarità basata sul coseno (***cosine similarity***), ovvero sul calcolo del coseno dell'angolo tra il vettore rappresentativo della query q ed il vettore rappresentativo del documento dj.

$$cosine(\mathbf{d}_j, \mathbf{q}) = \frac{\langle \mathbf{d}_j \bullet \mathbf{q} \rangle}{\| \mathbf{d}_j \| \times \| \mathbf{q} \|} = \frac{\sum_{i=1}^{|V|} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{|V|} w_{ij}^2} \times \sqrt{\sum_{i=1}^{|V|} w_{iq}^2}}$$

La classificazione dei documenti viene determinata in base ai valori di similarità rispetto alla query espressa dall'utente, ossia calcolando la distanza (il coseno dell'angolo) tra la rappresentazione del documento e quella della query nello spazio vettoriale. I documenti più "vicini" alla query nello spazio vettoriale saranno considerati quelli maggiormente attinenti alla query.

Volendo esemplificare quanto fin qui espresso proponiamo nel seguito un esempio:

Lo spazio dei documenti è definito da 3 termini:
hardware, software, users

Un insieme di documenti è rappresentato come segue nello spazio vettoriale in base alla presenza o meno dei termini sopra indicati :

A1=(1, 0, 0),	**A2=(0, 1, 0),**	**A3=(0, 0, 1)**
A4=(1, 1, 0),	**A5=(1, 0, 1),**	**A6=(0, 1, 1)**
A7=(1, 1, 1),	**A8=(1, 0, 1),**	**A9=(0, 1, 1)**

Nel modello booleano:

- I documenti A4, A7 saranno recuperati in "AND"
- I documenti A1, A2, A4, A5, A6, A7, A8, A9 saranno recuperati in "OR"

Nel modello vettoriale:

- La query q è rappresentata come il vettore (1, 1, 0) nello spazio vettoriale
- Calcolando i valori di similarità basati sul calcolo del coseno abbiamo:

S(q, A1)=0.71,	**S(q, A2)=0.71,**	**S(q, A3)=0**
S(q, A4)=1	**S(q, A5)=0.5**	**S(q, A6)=0.5**
S(q, A7)=0.82	**S(q, A8)=0.5,**	**S(q, A9)=0.5**

Come è facile immaginare tale formula viene frequentemente adottata anche nei task di clustering dei documenti in cui ogni cluster è composto dai documenti maggiormente "simili" tra loro: anche in questi casi dunque la misurazione della similarità tra documenti viene calcolata utilizzando il metodo della ***cosine similarity*** (similarità basata sul calcolo del coseno dell'angolo tra le rappresentazioni vettoriali dei documenti).

Negli anni sono state tuttavia proposte varie formule di pesatura all'interno dei modelli vettoriali [15], alcune più efficaci di altre; inoltre, sono state implementate varie tecniche per definire le relazioni tra i termini[16]. La maggior parte di queste tecniche si basano sull'espansione delle rappresentazioni del documento attraverso una trasformazione lineare T: il vettore D_i diventa $T * D_i$. Generalmente, la matrice T rappresenta una matrice di similarità termine- termine, che può essere definita costruendo un thesaurus dato a priori, o creando automaticamente una matrice di similarità semantica. In particolare, nel Generalized Vector-Space Model (GVSM)[16] la matrice T è una matrice termine-documento (ossia la trasposizione della matrice documento- termine, di cui un esempio è indicato nella Tabella 1 del paragrafo 2.1.1.2). Un altro interessante approccio è rappresentato dal ***Latent Semantic Analysis (LSA).*** L'analisi semantica latente[17]

[15] Buckley, C., Singhal, A., Mitra, M., Salton, G.: New retrieval approaches using SMART. In Proceedings TREC-4, NIST publication #500-236, Gaithersburg (MD) (1996) 25-48

[16] Wong, S.K.M., Ziarko, W., Raghavan, V.V.: On modelling of information retrieval concepts in vector spaces. ACM - Transactions on Database Systems, 12 (1987) 723-730.

[17] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., Harshman, R.: Indexing by latent semantic

consente la derivazione automatica di informazioni semantiche (es.: sinonimia e polisemia) da un dato insieme di documenti attraverso l'analisi della co-occorrenza dei termini. In particolare, due termini che sono sinonimi gli uni degli altri avranno probabilmente lo stesso profilo (cioè righe simili) nella matrice termine-documento. Tali correlazioni sono evidenziate in LSA attraverso una decomposizione della matrice termine-documento in valori singolari (***Singular Value Decomposition***). Più precisamente, sia C la matrice termine-documento. Quindi, la decomposizione del valore singolare (SVD) di C mira a identificare due matrici ortogonali U e V e una matrice diagonale tale che:

$$\mathbf{C} = \mathbf{U}\,\Sigma\,\mathbf{V}^t$$

dove t indica sta ad indicare la trasposizione. Supponendo che gli autovalori siano organizzati in valori decrescenti, si può ridurre la dimensionalità mantenendo solo le prime k colonne di U ($\mathbf{U_k}$), e le prime k righe di V ($\mathbf{V_k}$) e la prima k elementi della diagonale Σ ($\mathbf{\Sigma_k}$). La matrice $\mathbf{U_k \Sigma_k V_k^t}$ può essere considerata come la più vicina approssimazione (secondo la norma di *Frobenius*) di C di rango k. Documenti e query possono pertanto essere proiettate nello spazio latente ottenuto (tramite U_k^t D) e direttamente confrontati, ad esempio utilizzando la formula del coseno già menzionata, in questo spazio. Studi scientifici hanno dimostrato come LSA ha fornito miglioramenti rispetto al modello standard dello spazio vettoriale su diversi insiemi di documenti, anche se le risorse richieste per il calcolo rappresentano ad oggi il principale svantaggio dell'approccio LSA.

2.2.4.3. *Modelli avanzati di IR*

Molto spesso le rappresentazioni dei documenti e delle query in IR possono essere imprecise tanto da portare a basse performance nella fasi di reperimento delle informazioni di interesse. Tale osservazione ha portato negli anni alla ricerca di metodologie basate sulla combinazione di varie rappresentazioni dei documenti [18] (es.: termini singoli, n-grams, frasi) o differenti strategie di ricerca (es.: differenti implementazioni di modelli vettoriali e/o modelli probabilistici)[19].

Queste strategie di combinazione, meglio note come fusione dei dati ("data fusion"), tendono nel complesso a migliorare le prestazioni dei sistemi di IR per 3 motivi fondamentali:

---

analysis. Journal of the American Society for Information Science, 41 (1990) 391-407.

[18]Savoy, J.: Combining multiple strategies for effective monolingual and cross-lingual retrieval. IR Journal, 7 (2004) 121-148

[19] Vogt, C.C., Cottrell, G.W.: Fusion via a linear combination of scores. IR Journal, 1 (1999) 151-173.

- Prima di tutto, vi è un processo di scrematura in cui vengono presi in considerazione solo i primi m elementi recuperati con i migliori score di classificazione; in questo caso è possibile combinare le risposte migliori ottenute dalle varie rappresentazioni dei documenti.
- In secondo luogo si può contare sull'effetto "coro", secondo cui se uno stesso documento viene reperito utilizzando differenti rappresentazioni quel documento ha maggiori probabilità di risultare rilevante.
- Infine, dobbiamo considerare che un modello di retrieval può fornire stime insolitamente alte in relazione alla rilevanza o meno di un documento. Un sistema combinato consente invece di reperire un maggior numero di documenti pertinenti fornendo uno score maggiore per quei documenti rilevanti che in base al modello singolo erano stati classificati come poco pertinenti migliorando di conseguenza le performance generali dell'intero sistema.

Questi approcci di "data fusion" hanno tuttavia l'inconveniente di richiedere un quantitativo maggiore di spazio di memoria e di tempi di processamento.

Un'altra area di ricerca finalizzata al miglioramento delle performance in ambito di IR è rappresentata dall'applicazione di tecniche per migliorare le stime di probabilità. Tali tecniche si basano sulla possibilità di ottenere feedback direttamente dagli utenti circa la pertinenza o meno dei documenti reperiti rispetto alle query. Nel Web, ad esempio, è risaputo che gli utenti mostrano maggior preferenza per le home page dei siti in quanto attraverso la home page è possibile accedere facilmente a tutti i servizi e le informazioni di interesse (es.: biglietti aerei, controllo degli orari, prenotazione di hotel, etc..). Come descritto in [20], possiamo considerare queste informazioni per assegnare una maggior rilevanza alle home page. Questa tecnica adoperata classicamente nei motori di ricerca può migliorare anche notevolmente le prestazioni di un sistema di IR. Inoltre la maggior parte dei motori di ricerca prende in considerazione i testi di ancoraggio ("anchor text"), ossia le sequenze di parole utilizzate per indicare la presenza di un hyperlink e che descrivono in modo compatto la pagina Web target. Ad esempio nel caso della home page della Microsoft, è possibile raccogliere tutti i testi di ancoraggio scritti da vari autori che puntano al sito www.microsoft.com. Tutte queste espressioni possono essere viste come parti di un insieme, costruito manualmente, formato da varianti terminologiche che indicizzano la pagina target (es.: "see Microsoft", "Microsoft", "Bill Gates Empire", etc..).

Altre metodologie per migliorare le performance di sistemi di IR sono infine basate sulle relazioni tra i documenti, che vengono valutate tramite specifiche misure quali l'accoppiamento bibliografico ("bibliographic coupling") che misura la similarità tra i documenti sulla base dei documenti riferiti nelle

[20] Kraaij, W., Westerveld, T., Hiemstra, D.: The importance of prior probabilities for entry page search. In Proceedings ACM-SIGIR'2002, Tempere (2002) 27{34.

bibliografie [21], o la "co-citation measure" utilizzata da Small [22] per costruire una rete di documenti: per ogni coppia di documenti, vengono contati gli articoli citati da entrambe.

Più recentemente i ricercatori hanno studiato la possibilità di utilizzare approcci basati sul **machine learning** in IR. Assumiamo, in tal senso, di avere una grosso insieme di documenti, un gran numero di query e di giudizi di pertinenza ("relevance judgement") di questi insiemi rispetto alle query. E' possibile vedere questi giudizi di pertinenza come un insieme di coppie (query, documento) annotate, in cui l'annotazione prende ad esempio valore 1 se il documento è rilevante per la query, e -1 se non lo è.

Al passo successivo è possibile trasformare una determinata coppia (query, documento) in un vettore che rappresenta un esempio. Tale trasformazione ha la finalità di astrarsi dai termini specifici presenti nella query e di solito viene costruita sulla base di una serie di features[23] quali:

- Numero di termini comuni tra la query e il documento;
- Media della rilevanza del termine nella collezione di documenti (average inverse term frequency), calcolata per ciascun termine presente nel documento;
- Larghezza della finestra in cui i termini della query appaiono nel documento;
- Calcolo della ***cosine similarity,*** tra la query e il documento;
- Misura probabilistica di rilevanza del documento;
- Valore PageRank del documento;
- il numero di link entranti/uscenti;

Le coppie (query, documento) vengono viste come vettori il processo termina con un task di classificazione binario, la cui finalità è quella di discriminare tra esempi positivi e negativi. Una volta che il classificatore sarà stato costruito, i documenti pertinenti rispetto ad una data query saranno quelli classificati nella classe positiva: ogni documento viene quindi accoppiato alla query, la coppia viene poi trasformata nella modalità sopra descritta per poi essere data in pasto al classificatore. In questo contesto possono essere adottati vari classificatori, ma normalmente viene preferito **SVM** (**Support Vector Machine**) che in vari contesti si è rivelato quello più efficace.

---

[21] Kessler, M.M.: Bibliographic coupling between scientic papers. American Documentation, 14 (1963) 10-25.

[22] Small, H.: Co-Citation in the scientic literature: A new measure of the relationship between two documents. Journal of the American Society for Information Science, 24 (1973) 265-269.

[23] Nallapati, R.: Discriminative models for information retrieval. In Proceedings ACM-SIGIR'2004, Sheeld (UK) (2004) 64-71.

Nallapati[23] , ricercatore scientifico in Machine Learning applicato al NLP e al Text Analytics, mostra come l'approccio basato sul machine learning migliora le prestazioni di un sistema IR standard quando sia il numero di features utilizzate nella rappresentazione vettoriale che il numero di giudizi di pertinenza sono sufficientemente vasti. E' inoltre possibile andare oltre la classificazione binaria e cercare di addestrare una funzione di ranking tramite la tecnica delle preferenze a coppie ("pairwise preferences") [24] e/o preferenze a liste ("listwise preferences")[25] .

Importante comunque evidenziare che l'elemento base del machine learning applicato ad IR, si basa su un insieme di giudizi di rilevanza. Potrebbe comunque capitare che solo alcune collezioni di documenti (ad esempio quelli presenti in TREC) potrebbero avere un insieme di giudizi di pertinenza associati, mentre la maggior parte delle collezioni di documenti, di fatto, non ce li hanno. La situazione cambia, invece, per il Web, in quanto la maggioranza dei motori di ricerca hanno accesso ai dati "clickthrough", ossia i dati che memorizzano tengono traccia che per una data query, un utente ha cliccato su un particolare documento proposto dal motore di ricerca, ; questi dati possono essere in parte utilizzati come annotazione per l'addestramento di modelli di machine learning.

Il campo della ricerca nell'ambito del machine learning applicato ad IR è allo stato attuale estremamente attivo e come vedremo più avanti questi approcci saranno alla base della proposta di Exprivia per lo sviluppo del componente Mediator.

2.2.5. *Misure di Valutazione di Information Retrieval*

Al fine di misurare le performance di un sistema di retrieval [26], è necessario prendere in considerazione la capacità del sistema di reperite informazioni/documenti pertinenti rispetto alla query immessa.

In alcuni casi, a fronte di una query, la risposta del sistema potrebbe essere unica o almeno il numero di risposte potrebbe essere abbastanza limitato, come ad esempio nel caso in cui si ricerca nel web una specifica home page.

In questi casi la misura di valutazione si baserà esclusivamente sul rango associato al primo documento reperito.

---

[24] Cao, Y., Xu, J., Liu, T;-Y., Li, H., Huang, Y., Hon, H.-W.: Adapting ranking SVM to document retrieval. In Proceedings of ACM-SIGIR'2006, Seattle (WA) (2006) 186-193

[25] Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., Li, H. Learning to rank: from pariwise approach to listwise approach. In Proceedings of ICML'2007, Corvalis (OR) (2007) 129-136.

[26] Buckley, C., Voorhees, E.M.: Retrieval system evaluation. In E.M. Voorhees, D.K. Harman (Eds): TREC. Experiment and Evaluation in Information Retrieval. The MIT Press, Cambridge (MA), (2005) 53-75.

Dal punto di vista operativo, per una data query q, se ***r*** fosse il rango del primo documento rilevante reperito dal sistema, il calcolo della performance della query sarebbe calcolato dalla formula ***1/r.*** Questo valore, denominato rango reciproco ("*reciprocal rank*" **RR**), varia tra 1 e 0, ottenendo 1 nel caso in cui il primo documento reperito risultasse pertinente, o 0 qualora non fossero reperite risposte corrette; se un documento rilevante venisse reperito come secondo documento anziché come primo, il valore di RR tenderebbe a diminuire.

Per misurare le performance di reperimento risultante a partire da un dato insieme di query, viene semplicemente calcolata la media su tutte le query. Tale valore, meglio noto come media del rango reciproco ("Mean Reciprocal Rank" MRR), viene utilizzato per determinare in che misura un motore di ricerca è in grado di reperire risposte corrette e presentarle tra i primi documenti reperiti.

In IR, solitamente, non si vuole misurare la capacità di un sistema di classificare un elemento pertinente, ma la capacità di estrarre tutte le informazioni rilevanti da un insieme di documenti.

In questo contesto è importante avere sia una elevata ***precision*** del sistema (frazione di documenti rilevanti tra quelli reperiti) che una elevata ***recall*** (frazione di documenti rilevanti che sono stati reperiti rispetto all'intero insieme di documenti rilevanti). Tali misure devono dunque rispondere alle domande: "Tutti i documenti reperiti sono rilevanti?" (per la precision) e "Sono stati reperiti tutti i documenti rilevanti?" (per la recall).

Al fine di ottenere una definizione che sintetizzi precision e recall viene generalmente utilizzata (soprattuto in ambito NLP) una espressione matematica basata sulla media armonica delle 2 misure denominata F1 o ***F-score*** [27]. Se denominassimo con *P* la precision e con *R* la recall, la misura ***F-score*** sarebbe data dalla seguente formula:

$$F1 = \frac{2 \cdot P \cdot R}{P + R}$$

In IR tuttavia è molto comune calcolare la precisione media (Average Precision AP) per ogni query misurando la precisione raggiunta in ogni elemento pertinente estratto e calcolando infine la media complessiva. Quindi per un dato set di query calcoliamo la media della precisione media (MAP), che varia tra 0 (non sono presenti elementi rilevanti) e 1 (tutte le voci pertinenti appaiono sempre nella parte superiore della lista di documenti reperiti).

[27] van Rijsbergen, C.J.: Information Retrieval. 2nd ed., Butterworths, London (UK), 1979.

Per meglio comprendere come viene calcolata la misura AP prendiamo in considerazione il seguente esempio relativo ad un task di riconoscimento delle immagini (object detection).

Nel nostro esempio supponiamo che l'intero dataset sia composto da un insieme di oggetti di cui 5 mele. Ordiniamo nella tabella che segue i primi 10 oggetti restituiti dall'algoritmo: il rango è mostrato in ordine crescente in base al livello di confidenza restituito dall'algoritmo (dal livello con più alta confidenza a quello con più bassa confidenza).

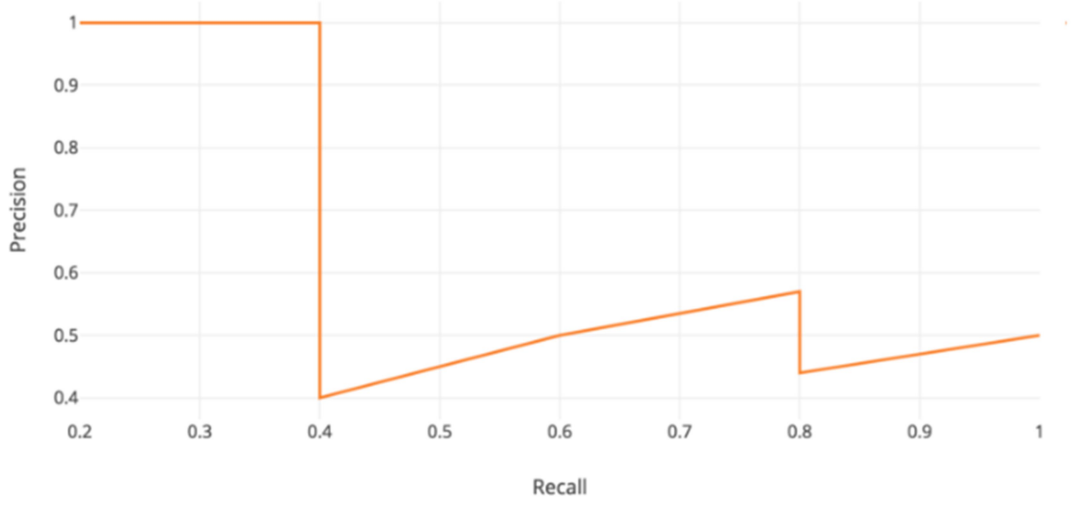
La seconda colonna indica se la classificazione dell'oggetto è corretta o meno, ossia se l'oggetto in questione raffigura effettivamente una mela.

Rank	Correct?	Precision	Recall
1	True	1.0	0.2
2	True	1.0	0.4
3	False	0.67	0.4
4	False	0.5	0.4
5	False	0.4	0.4
6	True	0.5	0.6
7	True	0.57	0.8
8	False	0.5	0.8
9	False	0.44	0.8
10	True	0.5	1.0

Prendendo in considerazione la riga con rango 3, la precision e la recall vengono così calcolate:

- La ***precision*** è data dai valori TP (True Positive) fino all'elemento con rango 3 e quindi ***Precision*** = 2/3 in quanto sono stati reperiti 2 risultati corretti su 3.
- La ***recall*** è data dalla percentuale degli elementi pertinenti reperiti rispetto a tutti quelli pertinenti nel dataset fino all'elemento di rango 3; pertanto nello specifico Recall = 2/5 in quanto sono 2 gli elementi pertinenti fino a quel momento su un totale di 5 elementi pertinenti.

I valori di recall crescono man mano che il rango cresce mentre la precision in questo caso presenta un andamento oscillante. Nel grafico sotto sono mostrati i valori della precision a fronte di quelli di recall.



Precision-recall curve

La misura di AP serve a delineare l'area sotto la curva di precision-recall; per calcolare la precisione media per ogni livello di recall è possibile utilizzare la seguente formula:

$$\overline{p}(r_i) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} p_j(r_i).$$

dove Q è l'insieme di tutte le query e $\mathbf{p_j(r_j)}$ rappresenta la precisione della query j al livello di recall r_j. Utilizzando la precisione media ad ogni livello di recall è possibile disegnare una curva precision-recall.

Jarvelin e Kekalainen hanno presentato [28] una misura di valutazione denominata Normalized Discounted Cumulative Gain (NDCG), che offre buoni risultati nelle situazioni in cui vengono valutati i giudizi di rilevanza, cioè quando essi comprendono più di due valori (rilevanti o meno). La premessa di NDCG è che i documenti altamente pertinenti che appaiono in una lista di risultati di ricerca dovrebbero essere penalizzati in quanto il valore di rilevanza graduale è proporzionalmente ridotto in forma logaritmica alla posizione del risultato.

NDCG calcola uno score di rilevanza su una lista di documenti attraverso il guadagno ottenuto da ogni singolo documento nella lista ridotto da un valore calcolato in base alla posizione in cui il documento appare. Questo punteggio viene quindi normalizzato rispetto al guadagno ottimale che può essere raggiunto, ottenendo un punteggio tra 0 e 1 (premesso che il valore più basso per la rilevanza è maggiore di 1).

[28] J arvelin, K., Kek al ainen, J.: Cumulated gain-based evaluation of IR techniques. Transaction on Information Systems, 20 (2002) 422{446.

Il punteggio NDCG per la lista costituita dai primi k documenti reperiti da un sistema IR è dato quindi dalla seguente formula:

$$N(k) = \overbrace{Z_k}^{\text{normalization}} \underbrace{\sum_{j=1}^{k}}_{\text{cumulative}} \overbrace{(2^{p(j)} - 1)}^{\text{gain}} \Big/ \underbrace{\log_2(j+1)}_{\text{position discount}}$$

in cui p_j corrisponde al valore di rilevanza del documento che appare alla posizione j.

Infine per poter determinare statisticamente se una determinata strategia di ricerca è meglio di un'altra, è possibile applicare test statistici[29] (Sign test, Wilcoxon signed ranks test, t-test o utilizzando la metodologia bootstrap).

2.3. Query Expansion e Relevance Feedback

Per fornire una migliore corrispondenza tra le esigenze di informazione degli utenti e i documenti, sono state suggerite tecniche di espansione della query. Il principio generale si basa sull'espansione della query mediante parole o frasi con significati simili o correlati a quelli che appaiono nella query originale, utilizzando le informazioni di un thesaurus[30][31] o ricavando queste informazioni dall'insieme di documenti. Per raggiungere questo obiettivo, gli approcci di espansione delle query si basano su:

- relazioni tra parole;
- meccanismi di selezione del termine;
- schemi di pesatura dei termini.

Le specifiche risposte a queste tre distinte soluzioni possono variare, portando ad un'ampia varietà di approcci di espansione della query[32].

Nel primo approccio, basato sulla ricerca di termini correlati tra loro, è possibile chiedere all'utente di selezionare i termini aggiuntivi da includere nella query espansa. Questo approccio può essere realizzato

---

[29] Conover, W.J.: Practical Nonparametric Statistics. 3rd edn. John Wiley & Sons, New York (NY) 1999.

[30] Voorhees, E.M.: Using WordNet to disambiguate word senses for text retrieval. In Proceedings ACM-SIGIR'93, Pittsburgh (PA) (1993) 171-180.

[31] Maisonnasse, L., Gaussier, E., Chevallet, J.P.: Multiplying concept sources for graph modeling. In C. Peters, V. Jijkoun, T. Mandl, H. Muller, D.W. Oard, A. Pe~nas, V. Petras, D. Santos, (Eds.): Advances in Multilingual and Multimodal Information Retrieval. LNCS #5152.Springer-Verlag, Berlin (2008) 585-592.

[32] Efthimiadis, E.N.: Query expansion. Annual Review of Information Science and Technology, 31 (1996) 121-187.

interattivamente attraverso la visualizzazione di un elenco classificato e ordinato per rilevanza degli elementi restituiti dalla query originaria.

Come seconda strategia, ***Rocchio***[33] propose di prendere in considerazione la pertinenza o meno dei primi n documenti reperiti dal sistema e ordinati per rilevanza, secondo indicazione manuale dall'utente. In questo caso, una nuova query verrà quindi creata automaticamente sotto forma di una combinazione lineare dei termini inclusi nella query precedente e i termini automaticamente estratti sia dai documenti ritenuti rilevanti dall'utente (e quindi con un peso positivo) sia da quelli non rilevanti (e quindi con peso negativo). Più precisamente, ogni nuova query viene derivata applicando la seguente formula:

$$w'_{qi} = \alpha \cdot w_{qi} + \beta \cdot \sum_{j=1}^{r} w_{ij} - \gamma \cdot \sum_{j=1}^{nr} w_{ij}$$

in cui w'_{qi} denota il peso associato al termine di query i-esimo, basato sul peso di questo termine nella query precedente (denotato da w_{qi}), e su w_{ij} che denota il peso del termini nei documenti che compaiono nelle prime k posizioni. Il valore r (rispettivamente nr) indica il numero di documenti rilevanti (e non rilevanti) che compaiono nel prime k posizioni. Le costanti positive α,β e ϒ sono fissate empiricamente. Studi empirici hanno tuttavia dimostrato che tale l'approccio è di solito abbastanza efficace.

Come terza tecnica, Buckley et al. [34] ha suggerito che anche senza guardali, si può presumere che i primi k documenti classificati siano rilevanti. Utilizzando questo approccio, è sufficiente impostare r= k e ϒ = 0 nell'equazione sopra espressa. Quest'ultimo metodo rappresenta in realtà una pseudo relevance-feedback e generalmente si dimostra efficace soprattutto nei casi in cui si gestiscono moli molto elevate di dati.

I metodi basati sulla relevance feedback si dimostrano molto efficaci nei casi in cui i risultati della query originale sono per lo più corretti, in altri casi le perfomance potrebbero anche diminuire.

Peat & Willet [35] in un loro studio forniscono una spiegazione associata a questo calo delle performance nell'utilizzo dei metodi di relevance feedback. Tale studio si basa sull'osservazione che i termini contenuti nella query hanno una frequenza maggiore rispetto agli altri termini. Pertanto l'espansione della query basata sulla co-occorrenza dei termini tenderanno ad includere termini che hanno una occorrenza elevata

---

[33] Rocchio, J.J.Jr.: Relevance feedback in information retrieval. In G. Salton (Ed.): The SMART Retrieval System. Prentice-Hall Inc., Englewood Cliffs (NJ) (1971) 313-323.

[34] Buckley, C., Singhal, A., Mitra, M., Salton, G.: New retrieval approaches using SMART. In Proceedings TREC-4, NIST publication #500-236, Gaithersburg (MD) (1996) 25-48.

[35] Peat, H.J., Willett, P.: The limitations of term co-occurrence data for query expansion in document retrieval systems. Journal of the American Society for Information Science, 42 (1991) 378-383.

all'interno dei documenti. In questi casi l'aggiunta di questi termini non porterà a discriminare tra documenti rilevanti e non-rilevanti e pertanto le performance in fase di reperimento saranno basse.

Sono stati eseguiti vari studi negli anni finalizzati al miglioramento delle tecniche basate sulla pseudo-relevance feedback. Il particolare in Kwok et al [36] viene proposto un approccio basato sul miglioramento della richiesta utilizzando il Web per ricercare termini relativi alle parole chiave contenute nella query originale.

2.3.1. *Applicazioni di NLP in IR*

Alla base della maggior parte delle lingue naturali ci sono le parole. Tuttavia, il termine "parole" è ambiguo ed è necessario essere più precisi nel distinguere tra la forma superficiale delle parole (ad esempio "horses") corrispondente ai token e il tipo di parola o lemma (voce nel dizionario, come "cavallo" nel nostro caso).

Inoltre, il significato specifico del termine "horse" (cavallo) non è sempre un animale con quattro zampe come ad esempio nel termine "horse-fly" (insetti denominati "tabanidi") o in espressioni come "Trojan Horse" (che nell'ambito della sicurezza informatica indica un tipo di malware), "to work like an horse" (lavorare come un cavallo).

Nella progettazione di sistemi IR elettronici, il componente morfologico[37] gioca un ruolo importante. Per la maggior parte dei sistemi di ricerca, come per la maggior parte dei linguaggi umane, le parole formano le unità di base per costruire frasi, espressioni e sentenze utilizzate per esprimere uno specifico significato. L'elaborazione appropriata di queste entità è quindi importante per migliorare le prestazioni nel reperimento delle informazioni di interesse promuovendo una corrispondenza pertinente senso-parola.

Inoltre, il modo in cui le parole si combinano e i significati che le parole esprimono sono cruciali per comprendere il contenuto di un documento e per rappresentare questo contenuto in modo accurato.

In questa sezione, esamineremo alcune delle principali relazioni tra IR e NLP. Lo faremo esaminando gli elementi fondamentali dell'Elaborazione del linguaggio naturale (morfologia, sintassi, semantica) nonché il ruolo che essi svolgono in ambito IR. Infine, forniremo una breve descrizione delle varie applicazioni in cui sono presenti connessioni dirette tra NLP e IR.

---

[36] Kwok K.L., Gruneld, L, Sun, H.L., Deng, P.: TREC2004 robust track experiments using PIRCS. In Proceedings TREC 2004, NIST publication #500-261, Gaithersburg (MD) (2005)

[37] Sproat, R.: Morphology and Computation. The MIT Press, Cambridge (MA), 1992.

2.3.1.1. *Analisi morfologica*

Come accennato in precedenza, l'obiettivo dell'analisi morfologica in IR è quello di far confluire varianti morfologiche nella stessa forma. In alcuni casi, viene eseguita solo ***un'analisi morfologica flessiva***, cioè un'analisi che riguarda la flessione dei nomi, aggettivi e verbi mediante aggiunta alla base lessicale di morfemi flessivi o desinenze.

L'analisi morfologica può essere ottenuta anche attraverso un'***analisi morfologica derivativa***, che riguarda la formazione di parole mediante l'aggiunta alla base lessicale di morfemi derivativi o affissi. Solitamente basata solo sull'aggiunta di suffissi, poiché i prefissi tendono a modificare radicalmente il significato di una parola (indicizzare ad esempio le due forme "decompose" e "recompose" come stesso token in IR non ha nessun senso).

Nella maggioranza dei casi l'analisi morfologica flessiva e derivativa vengono utilizzate congiuntamente.

Mentre gli analizzatori flessivi, basati su lessici elettronici, esistono in molti linguaggi non si può dire la stessa cosa degli analizzatori derivazionali; inoltre la comunità IR si è affidata negli anni a strumenti basati sull'identificazione della radice della parola ("word stem"), strumenti denominati in gergo tecnico "stemmers", che saranno brevemente descritti nel seguito.

Un algoritmo di stemming può consistere semplicemente nella rimozione dei suffissi flessivi, mentre per rimuovere invece i suffissi derivativi (es.: "-ment", "-ably", "-ship" nella lingua inglese) sono stati proposti approcci molto più sofisticati. Per l'inglese tale procedura consiste nel far confluire forme singolari e plurali ("car" e "cars") o nel rimuovere il participio passato terminante in "-ed" e il gerundio o participio presente terminanti in "-ing" ("eating" e "eat").

Questi algoritmi di stemming, basati esclusivamente sulla rimozione delle inflessioni morfologiche, sono detti "algoritmi leggeri di eliminazione del suffisso" e sono generalmente utilizzati per lavorare con testi generali in qualunque linguaggio. . Ad esempio lo stemmer di Lovins [38] è basato su una lista di oltre 260 suffissi, mentre l'algoritmo di Porter[39] si basa su circa 60 suffissi.

In questi casi la rimozione dei suffissi è controllata attraverso l'aggiunta di restrizioni quantitative (es.: "-ing" dovrebbe essere rimosso se lo stem risultante dalla rimozione ha più di 3 lettere come nella parola "running", ma non nella parola "king") o restrizioni qualitative (es.: "-ize" dovrebbe essere rimosso se lo stem risultante dalla rimozione non finisce per "e" come in "seize"). Inoltre possono essere applicate alcune regole di correzione di spelling ad hoc al fine di migliorare l'accuratezza della combinazione (es.: "running"

---

[38] Lovins, J.B.: Development of a stemming algorithm. Mechanical Translation and Computational Linguistics, 11 (1968) 22{31.

[39] Porter, M.F.: An algorithm for suffix stripping. Program, 14 (1980) 130-137.

diventa "run" e non "runn"). Di sicuro non devono essere sottoposti a processo di stemming i nomi propri come "Collins" o "Hawking" almeno nei casi in cui il sistema riesce a riconoscerli.

Alcune specifiche procedure di stemming sono indicate per domini specifici (es.: la medicina) o specifiche collezioni di documenti come quelli di Xu e Croft [40] che hanno suggerito l'utilizzo di algoritmi di stemming basati su un approccio "corpus-based" che riflette maggiormente il linguaggio utilizzato, includendo la frequenza delle parole e altre statistiche basate sulla co-occorrenza.

Un aspetto negativo delle procedure di stemming è legato al fatto che esse non tengono conto del significato delle parole e quindi tendono a commettere errori: tali errori possono essere dovuti ad un "over-stemming" (es. ."general" diventa "gener" e "organization" diventa "organ") o ad un "upper-stemming" (ad esempio nello stemmer di Porter le parole "create" e "creation" non confluiscono nella stessa radice). Per tali motivi è stato suggerito l'utilizzo di un dizionario on-line al fine di migliorare il raggruppamento dei termini.[41] [42]

Lo sviluppo di un analizzatore morfologico dipende molto dal linguaggio utilizzato. La morfologia flessiva in inglese è relativamente semplice: le forme plurale sono classicamente denotate mediante aggiunta di una "s" finale (con alcune eccezione come "feet" e "foot"), mentre le forme femminili sono ottenute mediante aggiunta di suffissi (es.: "actor" e "actress") e inoltre non è necessario rilevare la concordanza tra nomi e aggettivi. Per costruire nuove parole (costruzione derivativa) possiamo aggiungere prefissi ("pay", "prepay") e/o suffissi ("bank", "banking", "banker"). Sul mercato sono presenti numerosi tool per identificare varianti morfologiche in differenti linguaggi, alcuni basati su dizionari on-line e analisi morfologiche standard (es.: https://dictionary.cambridge.org/it/) o su procedure di stemming dedicate a differenti linguaggi (es.: http://snowball.tartarus.org).

2.3.1.2. *Analisi sintattica*

La maggior parte dei sistemi IR indicizza i documenti sulla base delle semplici parole che contengono. Tuttavia, ci sono stati molti tentativi finalizzati ad utilizzare indici più complessi, comprendenti più termini, al fine di ottenere una descrizione più precisa del contenuto dei documenti.

---

[40] Xu, J., Croft, B.W.: Corpus-based stemming using cooccurrence of word variants. ACM - Transactions on Information Systems, 16 (1998) 61{81.

[41] Krovetz, R., Croft, B.W.: Lexical ambiguity and information retrieval. ACM -Transactions on Information Systems, 10 (1992) 115-141.

[42] Savoy, J.: Stemming of French words based on grammatical category. Journal of the American Society for Information Science, 44 (1993) 1-9.

Uno dei primi tentativi, indicato come "coppie adiacenti" (descritte in [43]), considera termini complessi costituiti dal contenuto di due parole adiacenti. Questo approccio è particolarmente indicato per la lingua inglese, in quanto consente di acquisire elementi terminologicamente consistenti e costituiti da sequenze Aggettivo Nome (**Adjective Noun**) o Nome Nome (**Noun Noun**) (e le loro combinazioni). Per altre lingue, in cui le composizioni terminologiche si formano ad esempio attraverso le preposizioni, possono essere utilizzate semplici espressioni regolari su tag part-of-speech per identificare elementi terminologici e aggiungerli in seguito ai termini dell'indice.

Più in generale, un analizzatore sintattico può essere utilizzato per identificare dipendenze a lunga-distanza tra parole (ossia dipendenze tra parole non vicine tra loro), in modo da migliorare l'indicizzazione dei documenti. I primi lavori in questa direzione erano basati su grammatiche "tradizionali", in cui viene effettuata la ricerca di una rappresentazione di una sentenza o frase. Fagan[44] indaga per esempio sull'impatto di un parser sintattico sulle prestazioni di un sistema IR.

I progressi nell'ambito del parsing superficiale (o leggero) a metà degli anni '90 portarono a un'importante miglioramento in questo genere di ambiti; tali parser sono infatti stati parzialmente definiti come generali e robusti. Hull[45] propone, ad esempio, di utilizzare un parser superficiale per identificare relazioni tra nomi complessi e verbi delle frasi. Coppie di termini così estratti (ad esempio coppie soggetto-verbo o coppie aggettivo-nome) vengono quindi aggiunti a parole semplici per generare un indice più ricco per i documenti. Studi condotti sul modello dello spazio vettoriale (Vector space Model) hanno mostrato ad esempio un leggero miglioramento sulla collezione TREC-5, quinta edizione della TREC. La Text REtrieval Conference (TREC), è una co-sponsorizzazione del "National Institute of Standards and Technology" (NIST) e "U.S. Department of Defense", iniziata nel 1992 come parte di "TIPSTER Text program". Suo scopo era di supportare la ricerca all'interno della comunità di information retrieval fornendo le infrastrutture necessarie per la valutazione su vasta scala di metodologie per il reperimento di testi.

Uno dei principali problemi relativi all'uso delle informazioni sintattiche è che i miglioramenti nelle prestazioni in IR sono stati limitati e altamente dipendenti dalla collezione interrogata.

---

[43] Salton, G., Yang, C.S., Yu, C.T.: A theory of term importance in automatic text analysis. Journal of the American Society for Information Science, 26 (1975) 33{44.

[44] Fagan, J.L.: The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. Journal of the American Society for Information Science, 40 (1989) 115-132.

[45] Hull, D.A., Grefenstette, G., Sch ultze, B.M., Gaussier, E., Sch utze, H., Pedersen, J.O.: Xerox TREC-5 site report: Routing, filtering, NLP, and Spanish track. In Proceedings TREC-5, NIST publication #500-238 Gaithersburg (MD) (1997) 167-180

I termini complessi acquisiti attraverso l'analisi sintattica possono essere visti come informazioni aggiuntive che possono aiutare a rifinire i risultati della query e quindi portare ad un incremento della precisione.
Tuttavia, in molti casi, i vari elementi dei termini complessi vengono presi indipendentemente l'uno dall'altro e nel complesso questo approccio non modifica in modo sostanzialmente le performance del sistema. Ad esempio, è molto probabile che un documento indicizzato con i due termini "information" e "retrieval" tratti di "information retrieval" per cui l'aggiunta di quest'ultimo combinazione in questo caso specifico non aggiunge alcuna nuova informazione.

2.3.1.3. *Analisi semantica*

L'analisi sintattica fornisce ulteriori dimensioni di indicizzazione, ma non affronta tuttavia il problema della mancata corrispondenza di vocabolario tra query e documenti. Un modo per affrontare questo problema è ricorrere a un **analizzatore semantico** al fine di sostituire i termini dell'indice con i relativi concetti che essi esprimono ed eseguire il matching tra query e documenti a un livello più astratto.
In assenza di sistemi robusti che forniscano un'analisi semantica completa delle frasi, la maggior parte degli studi in ambito IR e semantica si sono concentrati sulla semantica lessicale e sulla possibilità di sostituire i termini indice standard con concetti o sensi delle parole (word sense).
Mentre alcuni studi hanno dimostrato che le procedure di ***word sense disambiguation*** possono essere utili in alcuni casi[46][47] , la maggioranza di essi ha cercato di fare affidamento su risorse semantiche già esistenti al fine di indicizzare sia i documenti che le query al livello concettuale.
Questi lavori hanno utilizzato risorse semantiche esistenti, sia generiche come ***Wordnet***[48], o specializzati per collezioni specifiche (ad esempio **UMLS**).
L'utilità dell'indicizzazione concettuale in domini specifici è stata mostrata in diversi studi, principalmente nel campo medico in cui sono presenti thesauri di grande copertura e reti semantiche. Ad esempio, i sistemi con le migliori prestazioni sul testo nell'attività ImageCLEFmed di CLEF [49] utilizzano metodi di indicizzazione

---

[46] Sanderson, M.: Word sense disambiguation and information retrieval. In Proceedings ACM-SIGIR'94, Dublin (1994) 142-151

[47] Schutze, H., Pedersen, J.O.: Information retrieval based on word senses. In Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval, Las Vegas (NV) (1995) 161-175.

[48] Voorhees, E.M.: Using WordNet™ to disambiguate word senses for text retrieval. In Proceedings ACM-SIGIR'93, Pittsburgh (PA) (1993) 171-180.

[49] Lacoste, C., Chevallet, J.P., Lim, J.-H., Wei, X., Raccoceanu, D., Le, T.H.D., Teodorescu, R., Vuillenemot, N.: Inter-media concept-based medical image indexing and retrieval with UMLS at IPAL. In C. Peters, P. Clough, F.C. Gey, J. Karlgren, B. Magnini, D.W. Oard, M. de Rijke, M. Stempfhuber (Eds): Evaluation of Multilingual

concettuale basati su modelli vettoriali o modelli linguistici. Nel TREC Genomics Track , Zhou et al.[50] ha utilizzato il thesaurus MeSH (Medical Subject Headings) e i databases Entrez per selezionare termini da pubblicazioni mediche. I termini nei documenti e le query sono state ampliate con le loro varianti trovate in queste risorse che sono state aggiunte per una migliore indicizzazione. E' stato dimostrato che questa tecnica ha portato ad un miglioramento significativo nei metodi di indicizzazione standard.

Altri ricercatori hanno cercato di andare oltre l'utilizzo dei concetti esplorando anche le relazioni tra essi. Vintar et al. [51] valuta l'utilità dei concetti UMLS e delle relazioni semantiche in ambito medico. Inizialmente vengono estratti concetti e relazioni tra documenti e query. Per selezionare le relazioni in una frase, vengono considerati due presupposti: (1) relazioni interessanti avvengono tra concetti interessanti; (2) le relazioni sono espresse mediante tipici indicatori lessicali come i verbi. Tuttavia, gli esperimenti effettuanti con un modello di spazio vettoriale mostrano che l'utilizzo di entrambi sia dei concetti che delle relazioni riducono le prestazioni ottenute con i soli concetti. Una linea simile di sviluppo si basa invece sull'approccio di modellizzazione linguistica applicato ad IR. L'approccio di modellazione linguistica in IR è stato proposto per la prima volta in[52], e esteso in[53]. Studi come (ad es. [54] [55]) hanno proposto la combinazione di modelli unigram con modelli bigram. I modelli unigram valutano la probabilità della co-occorrenza di "n" parole come:

$$P_{\text{uni}}(t_1 t_2 t_3) = P(t_1)P(t_2)P(t_3)$$

imponendo l'indipendenza della presenza delle varie parole in un testo, assunzione che genericamente nelle lingue non è vero; i modelli bigram invece modificano l'assunzione imponendo che l'ordine in cui le parole compaiono in un testo sono importanti e valutando quindi la probabilità come di seguito riportato:

---

and Multi-modal Information Retrieval. LNCS #4730. Springer-Verlag, Berlin (2007) 694-701.

[50] Zhou, W., Yu, C., Smalheiser, N., Torvik, V., Hong, J.: Knowledge-intensive conceptual retrieval and passage extraction of biomedical literature. In Proceedings ACM-SIGIR'2007, Amsterdam (2007) 655-662.

[51] Vintar, S., Buitelaar, P., Volk, M.: Semantic relations in concept-based crosslanguage medical information retrieval. In Proceedings of the ECML/PKDD Workshop on Adaptive Text Extraction and Mining (ATEM), Catvat{Dubrovnik (2003).

[52] Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In Proceedings ACM-SIGIR'98, Melbourne (1998) 275-281.

[53] Hiemstra, D.: Using Language Models for Information Retrieval. PhD. Thesis (2000).

[54] Srikanth, M., Srikanth, R.: Biterm language models for document retrieval. In Proceedings ACM-SIGIR'2002, Tempere (2002) 425-426

[55] Song, F., Croft, W.B.: A general language model for information retrieval. In Proceedings ACM-CIKM'99, Kensas City (MI) (1999) 316-321.

$$P(w_1, w_2, \cdots, w_n) = P(w_1)P(w_2|w_1)\cdots P(w_n|w_{n-1})$$

Altri studi hanno esteso il modello per affrontare le dipendenze sintattiche. Più recentemente, Maisonnasse et al. [56] hanno proposto una generalizzazione di questi approcci proponendo le ***reti semantiche***: tale approccio si basa sull'osservazione che qualsiasi grafico aciclico diretto può essere utilizzato per rappresentare documenti e query.

Quest'ultimo approccio è stato applicato in ambito medico, dove i documenti sono indicizzati a livello concettuale con UMLS e le relazioni tra concetti vengono aggiunte se due concetti che co-occorrono nella stessa frase sono presenti nella rete semantica associata a UMLS. I risultati mostrano che l'indicizzazione concettuale produce un miglioramento significativo rispetto all'indicizzazione standard in ambito IR; inoltre l'uso di relazioni semantiche migliora ulteriormente la precisione del sistema.

Questi ultimi risultati mostrano che, a condizione che si abbia a disposizione risorse semantiche adatte alla collezione (come è UMLS per il dominio medico), vi sono significativi vantaggi nell'utilizzo dell'indicizzazione semantica.

2.3.1.4. *Esempi di applicazioni di NLP in IR*

Esistono numerose applicazioni basate su modelli e metodi di IR ed NLP; di seguito alcuni esempi:

- La categorizzazione del testo (**Text Categorization**) è ad esempio un ambito in cui sono state condotte ricerca in entrambe le comunità, IR e NLP (oltre a quelle del Machine Learning e del Data Mining). L'obiettivo della categorizzazione del testo è l'assegnazione automatica di nuovi documenti a categorie esistenti. La maggior parte degli approcci sono attualmente basati sul machine learning, in cui documenti già classificati vengono utilizzati per l'addestramento del sistema. I documenti vengono rappresentati secondo il modello dello spazio vettoriale (Vector Space Model) a volte arricchiti con l'aggiunta di procedure successive, basate su **named entity recognition** e **term extraction**.
- Una seconda applicazione in cui vengono utilizzati i metodi sia da IR che da NLP è la document summarization, che mira a fornire una sintesi, in poche frasi, di un documento o di una raccolta di documenti. Gli approcci attuali si concentrano sull'estrazione di frasi chiave (a volte parti di frasi) dal documento o dalla raccolta di documenti e sulla loro visualizzazione in modo opportuno.
- Una terza applicazione è BioNLP, che si concentra sull'elaborazione dei documenti di testo nel dominio biologico. Per quanto riguarda il dominio medico, esistono diverse basi di conoscenza nel

[56] Maisonnasse, L., Gaussier, E., Chevallet, J.P.: Revisiting the dependence language model for information retrieval. In Proceedings ACM-SIGIR'2007, Amsterdam (2007) 695-696.

dominio biologico che possono essere utilizzate per rendere più accurata la rappresentazione dei documenti. Tuttavia, il tipo di informazioni ricercata dai biologi è complessa ed è necessario sviluppare una intera gamma di tecnologie per essere in grado di soddisfare le esigenze dei biologi.

- Un'altra importante applicazione in questi ambiti che merita di essere menzionata è il ***Question-Answering***, che mira a fornire risposte precise alle domande, operando al contrario di come avviene normalmente in IR, dove vengono fornite risposte su interi documenti o su interi paragrafi. Qui vengono fornite solo visione parziali di un documento e non la loro interezza. La maggior parte dei sistemi di question-answering si basano su una stretta combinazione di tecniche IR e NLP, portando a sistemi che integrano molte delle tecnologie esistenti di questi due domini.

2.4. Web Mining e IR

Le tecniche adottante nei sistemi di Information Retrieval possono essere combinate con tecniche di Web Mining (WM) al fine di estrapolare dalle pagine web la conoscenza utile a rispondere in maniera adeguata alle richieste dell'utente.

Il WM mira a trovare informazioni o conoscenze dalle strutture ipertestuali del web, dai contenuti della pagina e dai dati di utilizzo. In particolare su questi ultimi vengono applicati entrambi gli algoritmi di Data Mining (DM), basandosi sulle teorie di IR per scoprire le regolarità nell'ambito del web. Dovuta alla natura eterogenea, semi-strutturata e non strutturata dei dati del web, il WM utilizza le tecniche di DM, ma non è esattamente un'applicazione di DM.

In WM, inoltre, si affrontano i problemi causati dall'accesso di informazioni disponibili su Internet, tra i quali vi è quello della classificazione automatica di pagine web in base al loro contenuto testuale. I classificatori sono costruiti a partire da pagine rappresentative fornite da un singolo utente o da un gruppo di utenti durante l'attività di ricerca in rete.

Il processo di WM è simile al processo di DM e si differenzia per la collezione dei dati. Con gli algoritmi DM tradizionali, i dati sono già raccolti e salvati in una basi di dati. In WM, si considera che la collezione dei dati sia un passo fondamentale in particolare nell'ambito del **web structure mining** (dove si prova ad estrarre conoscenza dalla struttura di una pagina web) ed **web content mining** (in cui si cerca di estrarre informazioni o conoscenza dal contenuto di una pagina web), dove viene trattata l'elaborazione di un grande numero di pagine web. Una volta collezionati i dati, i passaggi successivi sono perlopiù analoghi rispetto a quelli del DM tradizionale.

2.4.1. *Tecniche per la raccolta di informazioni dal Web*

Per reperire le informazione dal WWW (World Wide Web) possono essere applicate diverse tecniche che possono essere elencate come segue:

- Web crawling, si analizzano i contenuti di una rete (o di un database) in un modo metodico e automatizzato, in genere per conto di un motore di ricerca. Nello specifico, un crawler è un programma automatizzato che acquisisce una copia dei contenuti di un sito andando ad analizzare ogni hyperlink presente all'interno delle pagine web e scaricare quindi ricorsivamente tutti i contenuti. I crawler solitamente possono fare uso di uno specifico file presente nei siti denominato robots.txt in cui sono riportate le restrizioni da applicare all'analisi delle pagine del sito. Non esiste un vero e proprio standard per il protocollo, ma si conviene che nel file ci siano N tuple ognuna espressa con la seguente sintassi "<campo> : <valore>" (es. User-agent: *). I più famosi crawlers sono: Google Crawler di Google, Scooter e Mercator per Altavista, ecc.
- Google dorking, tecnica che permette di interrogare in modo automatico un motore di ricerca tramite query per ottenere i risultati. Nato come metodo per trovare vulnerabilità dei sistemi sulla rete tramite l'utilizzo di Google ma che attualmente si può applicare anche ad altri motori di ricerca, quali ad esempio Bing e Shodan, e ad ambiti che esulano dalla sicurezza.
- RSS/ATOM, sono popolari formati che permettono ai siti di distribuire contenuti web. In questo caso è il sito che informa quali informazioni vuole esporre ed in quale categoria.
- Social, in questo caso non è un vero e proprio modo di reperire le informazioni sul web ma si utilizzano le API che i motori social mettono a disposizione. I contenuti social non sono di solito indicizzati dai motori di ricerca quali Google o Bing, questo perché i Social Network vedono nelle informazioni che loro possiedono un valore che non vogliono condividere facilmente con gli altri per non dare vantaggi ad altri che li possono sfruttare per generare introiti da pubblicità.
- Pastebin, come per i Social in questa categoria rientrano dei siti specifici che permettono di condividere contenuti testuali che non vengono indicizzati dai motori di ricerca. Tale modalità di condivisione è stato spesso un veicolo per attività illecite quali condivisione delle credenziali di accesso ai siti, così come le informazioni associate alle carte di credito di ignari possessori.

2.5. Descrizione tecniche specifiche

Il Mediator, componente oggetto di questa analisi, provvederà a mettere a disposizione un layer di connettori che accedono al web applicando alcune delle tecniche menzionate nella sezione precedente: RSS, Google Dork, Social e Crawler.

Le pagine web reperite dal layer di connettori, verranno annotate semanticamente per ricercare delle entità quali persone, aziende, ecc. Questa entità identificate e il resto dei contenuti testuali verranno indicizzati da un motore di ricerca (Media Index Client) che ne permetterà il successivo reperimento e visualizzazione.

Per lo "Scouting Tecnologico", ambito di utilizzo del Mediator, per ogni richiesta dell'utente verrà interrogato il motore di ricerca per ottenere tutte le evidenze ritenute pertinente limitando la ricerca a:

- Pubblicazioni scientifiche
    - https://www.scienceopen.com
    - https://loop.frontiersin.org
    - https://profiles.impactstory.org
    - https://mathscinet.ams.org/mathscinet
    - https://publons.com/
    - https://info.growkudos.com/
    - https://www.semanticscholar.org/
    - https://www.mendeley.com/
    - https://www.academia.edu/
    - https://academic.microsoft.com/home
- Brevetti registrati
    - Brevetti indicizzati da google: https://www.google.com/patents
    - Brevetti europei: https://www.epo.org
    - Brevetti americani: https://www.uspto.org

Per la determinazione del "Business Model", anch'esso ambito di utilizzo del Mediator, per ogni richiesta dell'utente verrò interrogato il motore di ricerca non effettuerà filtri sulle ricerche ma analizzerà le pagine indicizzate da Google, tenendo sempre presente le limitazioni del motore di ricerca, che consistono nella possibilità di avere a disposizione un massimo di 1000 risultati per ogni query espressa.

Per tutti e due i casi d'uso il Mediator, dopo aver reperito i documenti di interesse, procederà alla loro analisi per estrarre le entità che potrebbero risultare rilevanti. Per lo "Scouting Tecnologico" le entità ritenute rilevanti sono:

- Algoritmi.
- Codici di brevetto.
- Nomi di persone.
- Nomi di virus (utili per l'ambito medico).
- Paesi.
- Cause di morte (utili per l'ambito medico).
- Date.

Per aiutare la compilazione di un "Business Model" le entità ritenute rilevanti sono:

- Codici di brevetto.

- Nomi di persone.
- Nomi di organizzazioni o aziende.
- Paesi.
- Date.

Per il riconoscimento di queste entità bisogna determinare quali di queste sono dipendenti dal contesto e quali invece non lo sono. Per esempio una data ha un formato che la rende sempre riconoscibile e identificabile a prescindere in quale contesto venga utilizzato. Il nome di una persona o di una organizzazione possono invece essere ambigui se non si osserva il contesto. Come esempio si consideri la parola "Sante", può essere il nome di una persona ma usato ad inizio frase potrebbe essere aggettivo come negli esempi che seguono:

"Sante è andato a mangiare con i colleghi"

"Sante donne dice sempre mia moglie"

Per discernere tra i casi è necessario andare a controllare in quale contesto la parola "Sante" è utilizzato. Esistono delle risorse linguistiche che forniscono diversi esempi in cui vengono etichettate le parole secondo un tassonomia ben specifica. Queste risorse linguistiche però si limitano ad evidenziare le seguenti tipologie: Persone, Organizzazioni, Luoghi, Paesi. I parser linguistici forniscono dei modelli per il riconoscimento di queste entità, modelli creati utilizzando le risorse linguistiche, ma lasciano la possibilità di estendere il riconoscimento tramite la definizione di annotatori specifici.
Per gli scopi del mediator per le entità sono stati definiti degli annotatori specifici:

- Algoritmi, elenco di algoritmi maggiormente conosciuti ed utilizzati.
- Codici di brevetto, aderente agli standard: europei, americani e cinesi.
- Nomi di virus, elenco non esaustivo dei virus
- Cause di morte, elenco non esaustivo di malattie ed incidenti che possono causare la morte.
- Date, aderente ai formati più comuni di scrittura delle date.

3. Il Mediator

3.1. Descrizione generale della componente

Il Mediator, componente della piattaforma Digital Future che andremo a descrivere nel presente technical report, è finalizzato a fornire servizi trasversali ai tre pillar finalizzati al supporto su ricerca e valutazione di brevetti e pubblicazioni scientifiche e al supporto per lo sviluppo di iniziative tecnologiche e di business. Tutto ciò dovrà essere possibile fondamentalmente attraverso la definizione di attività specifiche di scouting rivolto verso tecnologie e di discovery intelligente delle varie informazioni; per tal motivo, questa componente la si può di fatto collocare nell'ambito tecnologico dell'*Information Retrieval*, tecnologia che oggi è centrale per la gestione, la rappresentazione, l'organizzazione e l'accesso alle informazioni diffuse nel web.

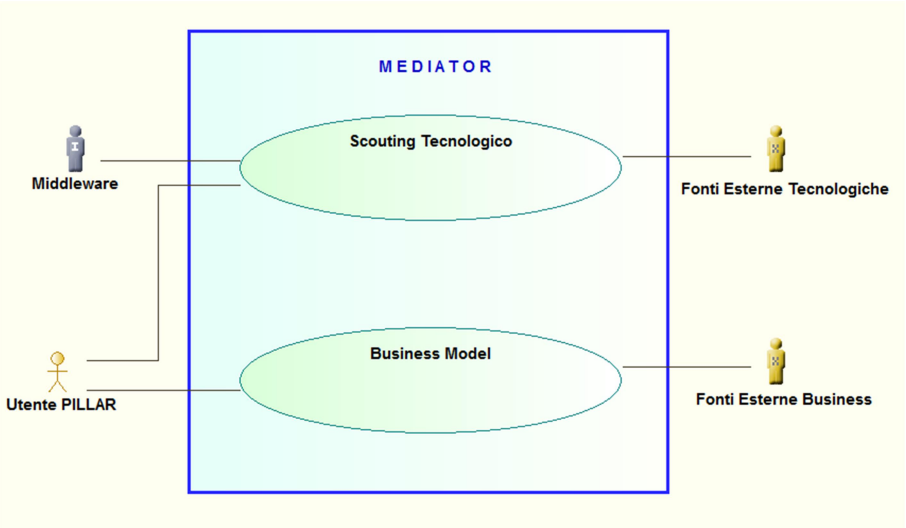


Figura 2 - Diagramma di contesto

Si presuppone, come mostrato nella figura precedente, che sia un utente pillar della piattaforma che il middleware stesso della piattaforma abbiano la possibilità di accedere al Mediator attraverso due funzionalità basi:

- Scouting tecnologico
- Business Model

3.1.1. *Le funzionalità*

Come menzionato precedentemente, compito del Mediator è quello di aiutare un esperto del settore a meglio focalizzare le proprie iniziative di business. Questo è ottenibile tramite l'applicazione di processi di

"scouting tecnologico" e di "analisi di mercato". Tali tipologie di processi è riferibile a processi di tipo OSINT (Open Source Intelligence), cioè il reperimento di informazioni da sorgenti aperte. Le tecniche che possono essere applicate in tale ambito sono varie, alcune delle quali rientrano nell'Intelligenza Artificiale che può essere più o meno invasiva nei processi. Nel particolare le tecniche per l'indicizzazione dei documenti reperiti tramite il processo di OSINT rientra nell'ambito di "Information Retrieval" dell'Intelligenza Artificiale, mentre la parte che ne analizza il contenuto e ne estrae informazioni prende il nome di "Information Extraction".

Nei sotto paragrafi verranno descritti più in dettaglio le varie funzioni che saranno esposte nel Mediator, definendo dettagliatamente le due principali funzionalità, "Scouting Tecnologico" "Business Model".

3.1.1.1. *Lo scouting tecnologico*

Il servizio di "technology scouting" consente di:

- Individuare le tecnologie e analizzare la letteratura scientifica di riferimento per risolvere uno specifico problema tecnologico.
- Valutare l'applicabilità di ciò che si è ottenuto precedentemente per un'idea innovativa di business.
- Identificare eventuali partner tecnologici e/o scientifici utili ad una efficace analisi di mercato, abilitando l'acquisizione di nuovo know-how.
- Scoprire principali competitor sul mercato in modo da meglio interpretare il settore di mercato su cui si vuole operare.

Per il primo punto è di cruciale importanza individuare quali siano le sorgenti affidabili da cui reperire le informazioni. Esistono in rete una moltitudine di piattaforme che permettono la raccolta di pubblicazione scientifiche, ma non tutte sono di uguale importanza perché dipendono fortemente dalla valenza sia delle pubblicazioni raccolte ma anche di quali conferenze sono interessate delle stesse. Anche la raccolta di brevetti è particolarmente importante anche se molte volte questi vengono descritti in modo molto generico rendendo difficile determinare quanto siano idee applicabili o puramente esercizi di stile. Per ottenere un elenco di sorgenti affidabili da cui reperire sia i brevetti che le pubblicazioni, ci si avvarrà di esperti. Alcune delle sorgenti che ci si aspetta di dover utilizzare, e che saranno forniti ed arricchiti dagli esperti, sono: United States Patent and Trademark Office (USPTO), European Patent Office (EPO), ResearchGate, ecc.

Il Mediator, fornirà anche informazioni a supporto del management, in modo da facilitare la redazione di business plan e analisi del mercato. Nel particolare l'identificazione di altre aziende, che possono essere o meno dei competitor, di istituti di ricerca, di università o di enti governativi, permetterà di delineare quali

siano gli interessi di mercato nel settore specifico di interesse. Importanti informazioni da estrarre saranno le quantità monetarie, se presenti, e le date per una eventuale introduzione di soluzioni sul mercato.

3.1.1.2. *Business Model*

La rivoluzione digitale iniziato negli anni '50 con il proliferare di computer e quindi la necessità di effettuare un passaggio da una logica analogica ad una digitale, ha portato le aziende a puntare sempre di più alla digitalizzazione del proprio patrimonio informativo. Questo cambiamento di paradigma che può essere annoverato come la terza rivoluzione industriale, ha abilitato negli ultimi hanno alla creazione di nuove modelli di business. Siamo in un periodo in cui c'è la corsa a creare nuove startup per emergere dalla massa delle aziende in modo differenziato. Per ottenere ciò non è però solo necessario essere presenti nel settore di mercato specifico, ma è necessario introdurre innovazione e determinare efficacemente quale segmento di mercato è idoneo per la propria nuova idea di business. La difficoltà nell'ottenimento di tali obiettivo fa si che solo poche startup riescono realmente ad emergere e pochissime riescono a ribaltare uno o più settori di mercato, basti considerare che secondo Fortune[57] nel 2015 solo 8 aziende nel mondo possono essere annoverate in questo elenco. Non sempre poi è necessario inventarsi nuovi modelli di business per portare vantaggi ad un'azienda ma copiare "modus operandi" consolidate su nuovi settori di mercato può essere la chiave vincente per fare business. E' di cruciale importanza quindi collezionare e analizzare ingenti quantità di dati che però possono essere di difficile e costosa gestione per un'azienda.
Recentemente, per invertire questa tendenza, si sta cercando di far evolvere il "dato" in un elemento, non solo di supporto al business, ma cardine in cui i dati vengono generalmente utilizzati per aggiungere un elemento alla "value proposition" del modello di business esistente.
Un elemento molto importante, sorgente di nuovi dati è la rete Internet, che si è sempre più radicato come strumento di condivisione di informazioni fornendo una relazione più diretta tra cliente ed azienda, come ad esempio nei siti di e-commerce, ma anche tra le aziende stesse. L'innovazione in questo caso può non solo essere quello di capire quali siano i competitor nel proprio settore di mercato ma di crearsi nuovi spazi di mercato da aggredire.
Il Mediator, deve quindi essere in grado di intercettare automaticamente da Internet il maggior numero di informazioni sui mercati di interesse condivisi con i competitor ma anche di far emergere possibili segmenti di mercato non pienamente espressi. In tale modo potrà essere più semplice per il business se è necessario ricercare nuove partnership, risorse o nuove attività in modo da estendere o integrare il proprio business, o al contrario definire un modello di business alternativo o parallelo a quelli già esistenti.

[57] http://fortune.com/2015/01/22/the-age-of-unicorns/

3.2. Architettura generale

Come descritto in §3.1.1.1, il Mediator deve permettere, a valle della sottomissione di una esigenza espressa da un utente esperto, di reperire da sorgenti esterne contenuti rilevanti. In Figura 3, è riportata parte dell'architettura di Digital Future che riguarda il componente Mediator. Questo dovrà reperire, dalle sorgenti identificate dagli esperti, i contenuti più rilevanti rispetto all'esigenza dell'utente. I documenti reperiti devono essere analizzati tramite i servizi di analisi messi a disposizione dal "Semantic & Ontological Layer" per estrarre tutte quelle entità ritenuti rilevanti per una corretta interpretazione dei contenuti. Il flusso può essere descritto genericamente come segue:

1. L'utente esprime una richiesta di un interesse a riguardo di una tecnologia e/o di un settore di mercato.
2. Il Mediator provvederà ad analizzare la richiesta espressa dall'utente ed incomincerà a reperire contenuti presenti dalle sorgenti che possono essere sia sorgenti web che database di contenuti.
3. I contenuti reperiti dalle sorgenti, saranno sottomesse al "Document Annotator" in modo da estrarre le informazioni (entità) di maggiore interesse.
4. All'utente vengono forniti i risultati del processo di ispezione.
5. L'utente può richiedere di visualizzare le relazioni che intercorrono tra i documenti e tra le entità ivi contenute.

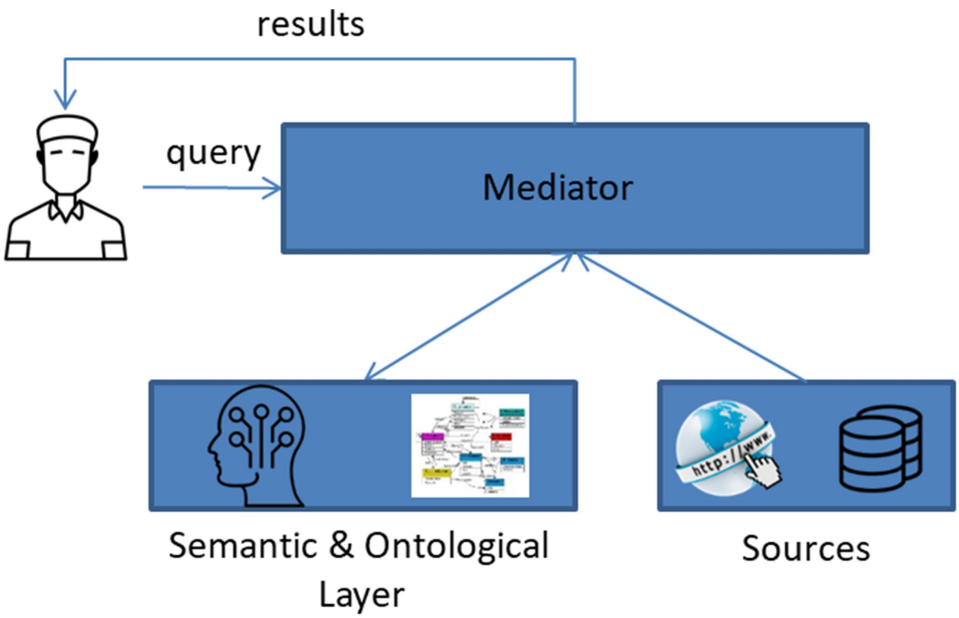


Figura 3 - Architettura di riferimento

Il Mediator internamente è strutturato a sua volta come riportato in Figura 4 - Architettura logica del Mediator con la sua architettura logica interna. Nel seguito verrà descritta la struttura interna del Mediator, dando risalto e descrizione ai componenti interni.

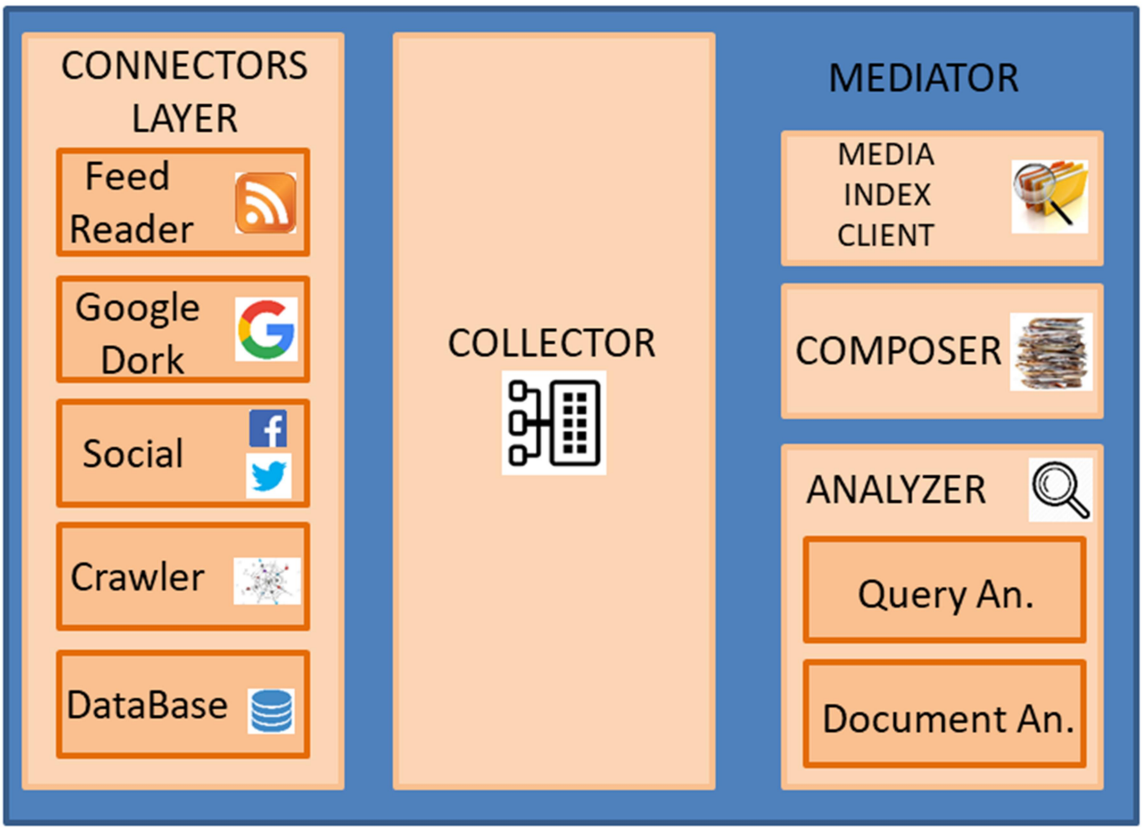


Figura 4 - Architettura logica del Mediator

Il "Connectors Layer" raccoglie tutti i componenti interni al Mediator, che hanno il compito di reperire le informazioni necessarie per soddisfare la richiesta espressa dall'utente. In questo componente sono presenti i seguenti componenti:

- *Feed Reader*, ha il compito di gestire tutte le interazioni con sorgenti di tipo "Feed", con formati di tipo RSS o Atom.
- *Google Dork*, permette di accedere all'indice di Google esprimendo richieste specifiche. L'utilizzo avviene tramite la sottomissione di una query nella sintassi supportata dal motore di ricerca di Google. Il motore di ricerca permette previa iscrizione ed ottenimento di credenziali un numero limitato di query che possono essere sottomesse ogni giorno.

- *Social*, consente di accedere ai contenuti pubblici pubblicati sulle piattaforme social Facebook e Twitter. Potranno essere reperiti i contenuti pubblicati da un utente che ricercate tramite attinenza ad una query di riferimento (modalità valida solo per Twitter).
- *Crawler*, permette di effettuare un download di tutti i contenuti presenti su una sito web (espresso sotto forma di URL). Il quantitativo e la tipologia di contenuti scaricabile, dipende dalla struttura del sito, nel caso di siti protetti i dinamici alcuni contenuti potrebbero non essere ottenibili. Inoltre la profondità di analisi ed il numero di contenuti da scaricare saranno parametri di configurazione da impostare.

Il componente *Media Index Client*, sottomette le informazioni reperite ed estratte al sistema di indicizzazione in modo che possano essere messe a disposizione in qualsiasi momento.

Il *Composer* permette, a valle della selezione di un utente di un elenco di documenti di interesse, comporre un diagramma che evidenzia le relazioni che intercorrono con le entità rilevate all'interno di essi.

Il componente *Analyzer* colleziona due componenti al suo interno:

- *Query Analyzer*, permette di analizzare semanticamente, utilizzando i servizi messi a disposizione dal "Semantic & Ontological Layer". Tale componente dovrà essere in grado di intercettare gli elementi richiesti dall'utente in modo da ottimizzare le richieste per il reperimento dei contenuti dalle sorgenti.
- *Document Analyzer*, in modo analogo al precedente esso analizza semanticamente i contenuti reperiti dalle sorgenti permettendo di estrarre elementi rilevanti presenti all'interno dei contenuti stessi, quali ad esempio: date, url, nomi di persone, aziende, ecc.

Infine il *Collector*, un componente che gestisce la sequenza di chiamate ai vari servizi esposti dal:

- "Connectors Layers".
- "Indexer".
- "Analyzer".

Nella successiva sezione verranno approfonditi in dettagli i componenti sopra menzionati, riportando l'interazione delle stesse.

3.2.1. *Descrizione delle componenti*

3.2.1.1. *Feed Reader*

Tale componente permette di accedere in modo schedulato ai contenuti pubblicati da una sorgente tramite Feed nei formati RSS o ATOM. Un Feed altro non è che un file di testo scritto nei formati menzionati precedentemente ed entrambi derivati dall'XML. Il componente allo scadere della sua schedulazione

accederà al feed, ne scaricherà il contenuto e lo analizzerà. Non tutti gli elementi del feed potranno essere di interesse, nel particolare gli elementi più utili saranno:

- il link associato al contenuto originale,
- il titolo associato al contenuto,
- la data di pubblicazione,
- l'abstract associato al contenuto,
- il contenuto.

Per sua natura, un feed può omettere dei campi, nel particolare sia l'abstract associato al contenuto che il contenuto stesso potranno essere più o meno esaustivi lasciando al link esterno la totalità del contenuto che si vuole pubblicare. Si andrà quindi prima ad analizzare l'abstract, epurandolo eventualmente di formattazioni HTML; nel caso in cui l'abstract è vuoto si andrà ad analizzare il contenuto nel feed epurandolo anche in questo caso di formattazioni HTML. Il link esterno verrà utilizzato per scaricare completamente il contenuto originale che verrà anch'esso analizzato in modo da epurarlo da formattazioni HTML ed eliminare parti irrilevanti quali banner pubblicitari o comunque poco rilevanti.

Al fine di evitare che il connettore effettui il download dello stesso contenuto più volte verrà implementato un meccanismo di "firma" del contenuto. Ad ogni esecuzione del connettore, se viene reperito un contenuto "simile" questo verrà ignorato; il concetto di similarità dovrà avere una soglia in modo da permettere un grado di libertà nella definizione di contenuto simile in modo che eventuali correzioni dello stesso vengano prese in considerazione, mentre cambiamenti minori quali aggiunta di punteggiatura vengano invece trascurati.

Come esempio dell'impostazione manuale di un connettore di tipo RSS sul canale di "Scienza e Tecnologia" di "Repubblica", si faccia riferimento a Figura 5.

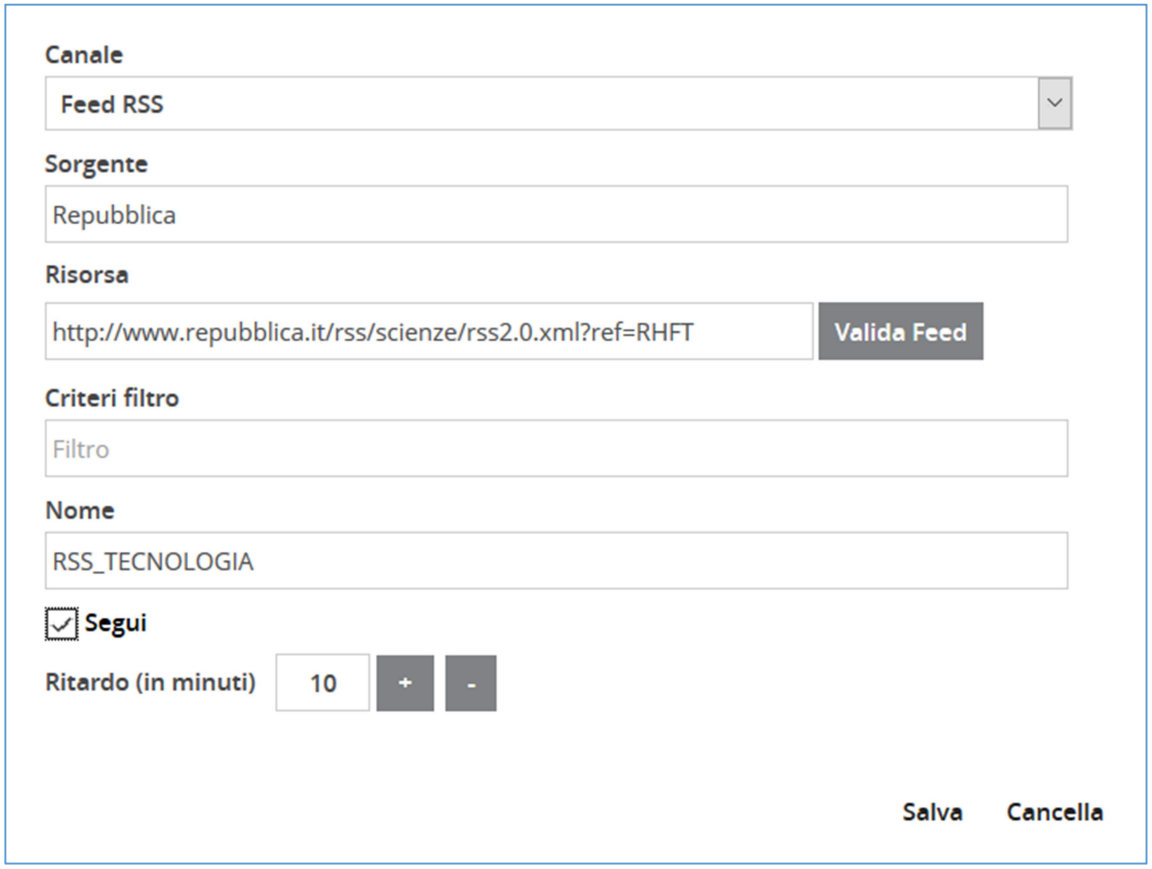


Figura 5 - Esempio connettore RSS

3.2.1.2. *Google Dork*

Tale componente permette di accedere in modo schedulato ai contenuti ricercabili sul motore di ricerca di Google, tramite la sottomissione di una query. Google mette a disposizione delle API denominate "Customsearch" che, previa registrazione, permette di eseguire ricerche sul suo indice fino ad un massimo di 100 giornaliere. Per sua natura di motore di ricerca gli aggiornamenti sull'indice non avvengono in modo così frequente da giustificare una schedulazione che contempli l'esecuzione per più volte al giorno. È presente la limitazione delle 100 query al giorno per uso delle API in modalità open, che saranno utilizzate nella fase di prototipazione della soluzione. In questa fase, per evitare situazioni vincolanti per il Mediator, si potrà ipotizzare di introdurre dei meccanismi che permetteranno di utilizzare in modo casuale più credenziali di accesso ai servizi di Google. Questo vincolo sarà oggetto comunque di valutazione in una eventuale successiva fase di industrializzazione, durante la quale si potrà procedere con una registrazione a pagamento per l'utilizzo dei servizi di Google. In tal modo il numero di sottomissioni giornaliere potranno aumentare a seconda delle esigenze,il cui costo, valutato ad oggi, potrebbe variare partendo da un minimo

di 5$ per 1.000 sottomissioni giornaliere e incrementare con l'aumentare del numero di sottomissioni, fino ad un massimo di 10.000 sottomissioni giornaliere.

Il servizio CustomSearch di Google altro non è che un servizio RESTFul con risposta in formato JSON. A valle della sottomissione di una richiesta opportunamente formattata, il JSON di risposta conterrà, in modo analogo ad un feed, con un formato specifico definito nello standard "OpenSearch v1.1"[58]. Gli elementi da estrarre dal JSON di risposta saranno:

- titolo del contenuto,
- url del link del contenuto,
- url epurato del protocollo,
- snippet (analogamente a come Google fa nell'intefaccia web).

La url associata al contenuto verrà utilizzata per scaricare il contenuto originale che verrà epurato da formattazioni HTML e di parti irrilevanti quali banner pubblicitari o comunque poco rilevanti.

Al fine di evitare che il connettore effettui il download dello stesso contenuto più volte verrà implementato un meccanismo di "firma" del contenuto. Ad ogni esecuzione del connettore, se viene reperito un contenuto "simile" questo verrà ignorato; il concetto di similarità dovrà avere una soglia in modo da permettere un grado di libertà nella definizione di contenuto simile in modo che eventuali correzioni dello stesso vengano prese in considerazione, mentre cambiamenti minori quali aggiunta di punteggiatura vengano invece trascurati.

Come esempio dell'impostazione manuale di un connettore di tipo "Google Dork" con ricerca su algoritmi di classificazione, si faccia riferimento a Figura 6.

[58] http://www.opensearch.org/Specifications/OpenSearch/1.1

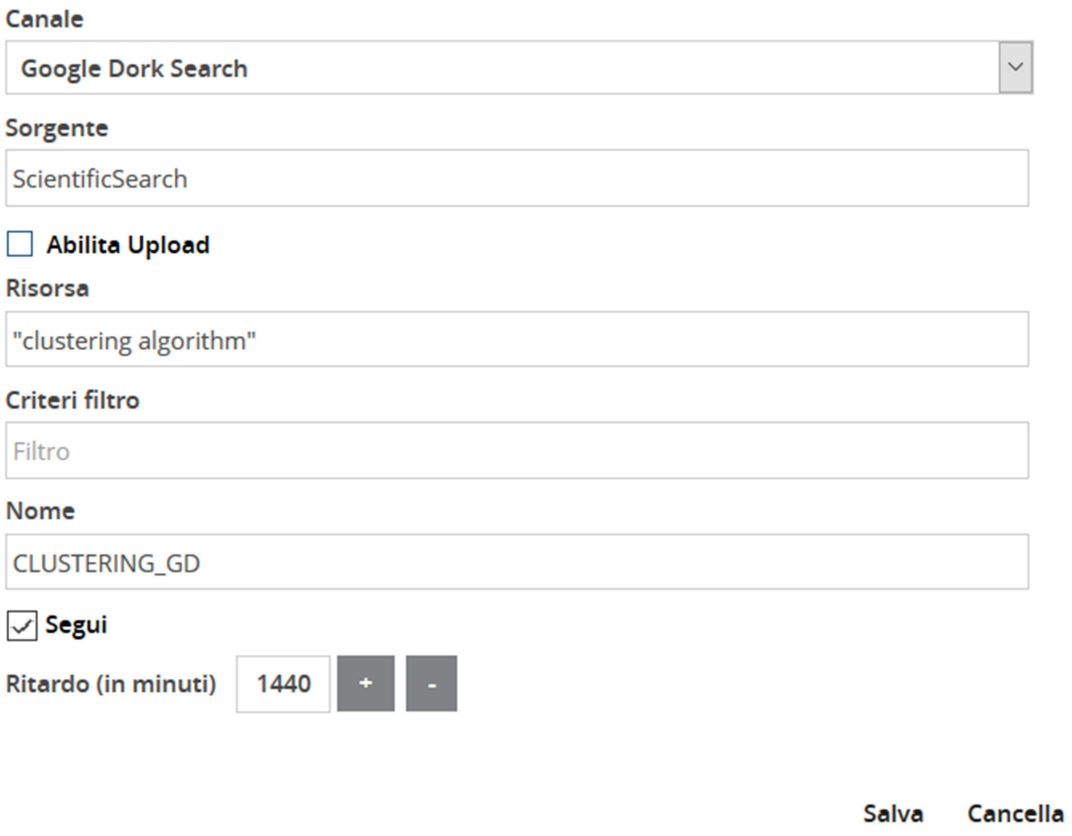


Figura 6 - Esempio connettore Google Dork

3.2.1.3. *Social*

Questo connettore permette di accedere ai social: Facebook e Twitter. Entrambe le piattaforme permettono l'accesso ai loro contenuti tramite API:

- Facebook, accesso ai post pubblici di una pagina Facebook.
- Twitter, accesso ai post di un profilo Twitter e ricerca sugli indici di Twitter.

Tali API utilizzano un meccanismo di tipo push-back, cioè dopo la sottomissione di una richiesta il canale di comunicazione rimane aperto in modo che il provider, Facebook o Twitter, forniscano in tempo reale i risultati. Per utilizzare tali servizi, si utilizzeranno librerie esterne per facilitare l'interazione con le api dei motori Social. A prescindere dalla tipologia di richiesta eseguita bisognerà estrarre il contenuto testuale, il titolo ed eventuali riferimenti esterni.

Al fine di evitare che il connettore effettui il download dello stesso contenuto più volte verrà implementato un meccanismo di "firma" del contenuto. Ad ogni esecuzione del connettore, se viene reperito un contenuto "simile" questo verrà ignorato; il concetto di similarità dovrà avere una soglia in modo da

permettere un grado di libertà nella definizione di contenuto simile in modo che eventuali correzioni dello stesso vengano prese in considerazione, mentre cambiamenti minori quali aggiunta di punteggiatura vengano invece trascurati.

Come esempio dell'impostazione manuale di un connettore di tipo "Twitter Search" con ricerca su algoritmi di classificazione, si faccia riferimento a Figura 7.

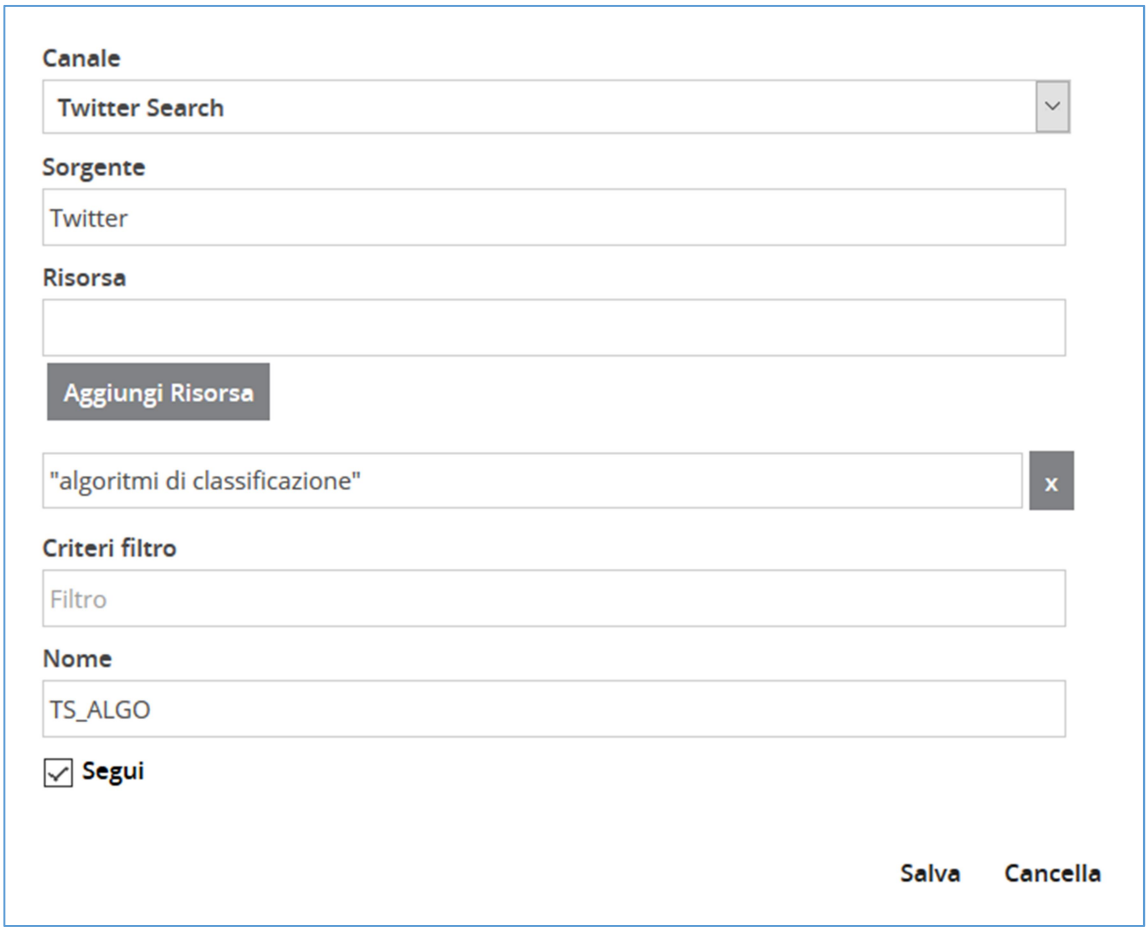


Figura 7 - Esempio connettore Twitter Search

3.2.1.4. *Crawler*

Tale componente permette di analizzare i contenuti di un sito web in modo automatizzato in modo analogo a come procede un motore di ricerca. A valle della definizione di una url, il componente procede con:

1. Scaricare la pagina web associata alla url.
2. Analizzare la pagina alla ricerca di tutti i collegamenti ipertestuali (se presenti).
3. Ripetere il passo 1 per tutti i collegamenti ipertestuali fino ad una clausola di terminazione.

La clausola di terminazione dovrà permettere di interrompere il processo al determinarsi di una delle seguenti condizioni:

- Raggiungimento del numero massimo di pagine scaricabili.
- Raggiungimento del numero massimo di livelli dalla pagina originale raggiunti.

Una corretta configurazione della clausola di terminazione è vitale per evitare di effettuare un download globale del sito saturando tutte le risorse disponibili dal Mediator.

Tutti contenuti scaricati dal Crawler verranno epurati da formattazioni HTML e di parti irrilevanti quali banner pubblicitari o comunque poco rilevanti. Inoltre, così come per altri connettori e al fine di evitare che il connettore effettui il download dello stesso contenuto più volte, verrà implementato un meccanismo di "firma" del contenuto. Ad ogni esecuzione del connettore, se viene reperito un contenuto "simile" questo verrà ignorato; il concetto di similarità dovrà avere una soglia in modo da permettere un grado di libertà nella definizione di contenuto simile in modo che eventuali correzioni dello stesso vengano prese in considerazione, mentre cambiamenti minori quali aggiunta di punteggiatura vengano invece trascurati.

3.2.1.5. *Indexer*

Questo componente permette di tenere traccia di tutti i contenuti reperiti tramite i connettori e di tutti i suoi metadati, quali ad esempio: titolo, data di pubblicazione, url, ecc.

L'indicizzazione dovrà avvenire tramite un motore di ricerca perché risulta essere lo strumento più adatto ogni qual volta sia necessario indicizzare contenuti testuali. In questi casi un DBMS risulta essere uno strumento poco efficiente. Per la natura dello strumento (Mediator) i contenuti e gli attributi potranno essere numerosi rendendo quindi di cruciale importanza la necessità di impiegare tecniche che tengano conto di concetti quali: replica dei dati, failover e load balancing. Esistono diversi motori di ricerca open source e commerciali che possono essere impiegati per lo scopo; la maggior parte fanno uso del motore di ricerca open source denominato Lucene[59]. Un ulteriore pregio di un motore di ricerca rispetto ad un DBMS risiede nella ampia disponibilità di modalità di interrogazione che primi hanno e che vanno oltre alla sintassi SQL supportata dai DBMS. E' possibile eseguire query:

- Di tipo booleano con operatori AND, OR e NOT.
- Fuzzy e per prossimità.
- Per intervalli di valori.
- Di similarità.

[59] https://lucene.apache.org/core/

3.2.1.6. *Composer*

Il Composer permette, a valle del processo di ricerca scatenato dalla richiesta di un utente, di generare un report che evidenzi come sono relazionate le entità rilevanti, per lo specifico dominio, nei documenti reperiti dalle sorgenti di riferimento. Come esempio si consideri Figura 8, dove posizionandosi sopra i nodi è possibile evidenziare le relazioni che intercorrono, come nell'esempio tra l'algoritmo "bayesian network" e le cause di morte: "infection" e "sepsis".

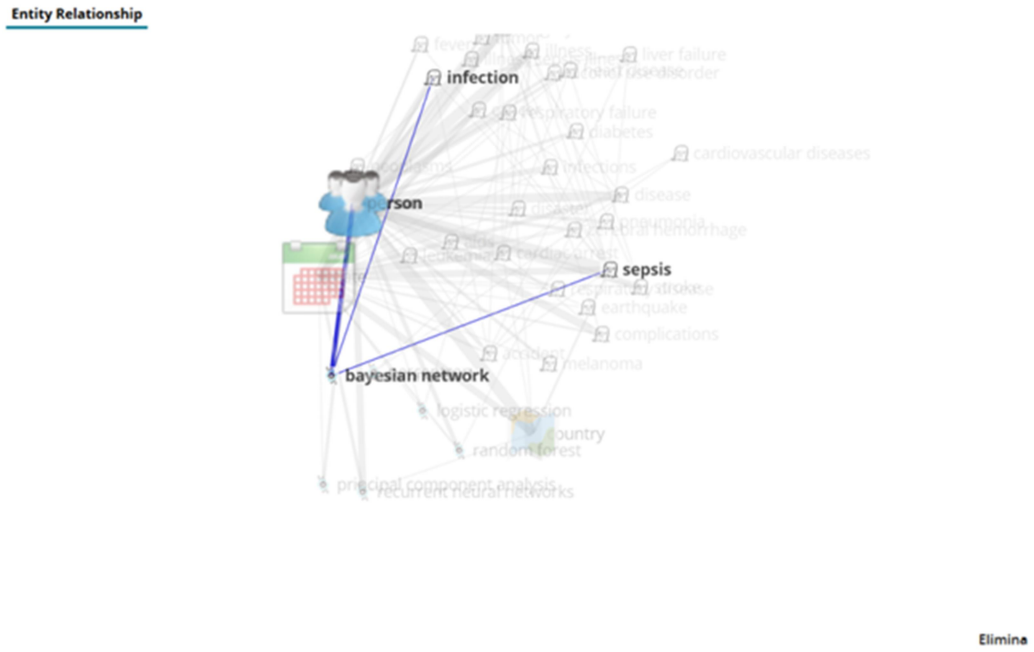


Figura 8 - Esempio output del composer, diagramma Entity Relationship

3.2.1.7. *Analyzer*

Nel layer "Analyzer" sono presenti quei componenti predisposti all'analisi linguistica delle richieste sottomesse dall'utente e i documenti reperiti nel processo di ispezione nelle sorgenti. Di seguito vengono riportate descrizioni associazione a tali componenti.

3.2.1.7.1. Query Analyzer

Il componente "Query Analyzer" procede con l'utilizzare tecniche di "Text Analysis" con lo scopo di meglio determinare quali siano i concetti espressi all'interno di una query. Per ottenere ciò esso opererà attraverso funzioni di analisi della query, utilizzando i servizi messi a disposizione dal "Semantic & Ontological Layer". Le informazioni da estrarre che risulteranno rilevanti per determinare i concetti espressi nella richiesta del cliente, saranno entità: organizzazioni, compagnie, quantità monetarie, date, ecc. Questi elementi possono

anche essere oggetto di espansione ontologica nel caso in cui questi siano presenti all'interno dell'ontologia caricata nel "Semantic & Ontological Layer" con una relazione similare alla sinonimia. L'output quindi fornito dal "Query Analyzer" è un insieme di elementi con le eventuali espansioni che possono essere utilizzare per reperire i contenuti nel processi di ispezione nelle sorgenti.

3.2.1.7.2. Document Analyzer

Analogamente al modulo precedente, il "Document Analyzer" procederà ad utilizzare i servizi messi a disposizione dal "Semantic & Ontological Layer" per estrarre tutte le informazioni presenti all'interno dei documenti reperiti dalle sorgenti che risultano essere rilevanti per una migliore comprensione dei contenuti. Si presuppone che le entità rilevanti siano similari a quelle estratte nel "Query Analyzer" e quindI: organizzaziono, compagnie, quantità monetarie, ecc.

3.2.1. *Integrazioni tra le componenti*

Tutti i componenti presenti all'interno del Mediator dovranno comunicare tra di loro tramite l'esposizione e l'interrogazione di servizi RESTFul. In tal modo sarà anche possibile instaurare logiche di resilienza e tolleranza ai guasti.

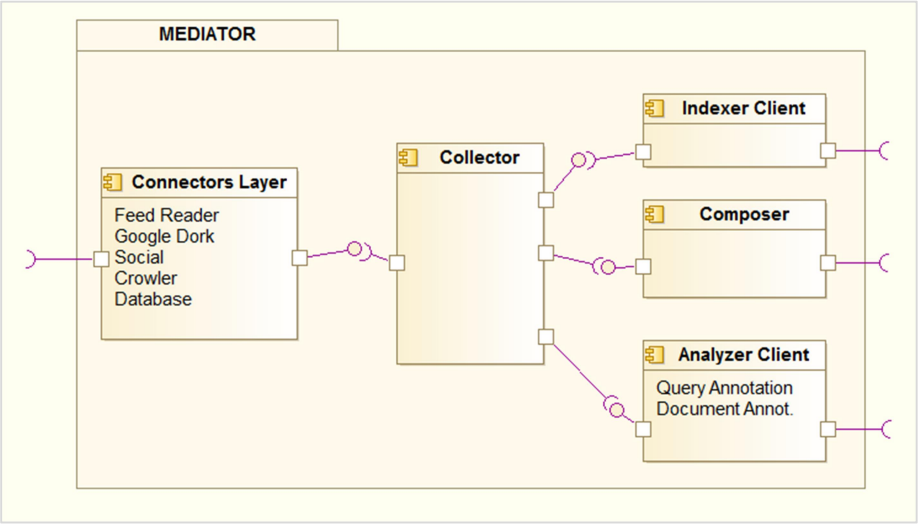


Figura 9 - Component Diagram

4. Bibliografia

[1] J. W. R. a. M. F. Henrik Brink, Real-World Machine Learning, September 2016.

[2] intelligenzaartificiale, «Machine Learning: l'apprendimento automatico,» [Online]. Available: http://www.intelligenzaartificiale.it/machine-learning/#Un_po8217_di_storia.

[3] A. Fogg, «Una storia di apprendimento profondo,» import.io, 30 maggio 2018. [Online]. Available: https://www.import.io/post/history-of-deep-learning/.

[4] «Machine Learning, storia di un algoritmo in evoluzione,» 26 04 2018. [Online]. Available: https://www.quix.it/blog/machine-learning/7095489.

[5] P. N. Sharkey, «Alan Turing: l'esperimento che ha modellato l'intelligenza artificiale,» 21 01 2012. [Online]. Available: https://www.bbc.com/news/technology-18475646.

[6] «First Robot 3D Mapping and Navigation in Ordinary Settings,» [Online]. Available: https://www.frc.ri.cmu.edu/~hpm/project.archive/robot.papers/2002/Commercial/1979.MRL.html.

[7] Tom-Mitchell, Machine-Learning, McGraw-Hill Science/Engineering/Math, March 1, 1997.

[8] A. Lui, «I Dato è Tratto: Machine Learning, cos'è e a cosa serve,» PMC International Service, 29 05 2017. [Online]. Available: https://www.youtube.com/watch?v=4S3a0JdPn1A.

[9] G. Gullo, «Cos'è il machine learning ?,» Profession AI, 11 02 2018. [Online]. Available: https://www.youtube.com/watch?v=UHx9tB2ObeY.

[10] Y. Mulkers, «Building the Machine Learning Infrastructure,» Jun 23 2017. [Online]. Available: https://medium.com/@YvesMulkers/building-the-machine-learning-infrastructure-f30336a9fc37.

[11] S. K. Jaiswal, «Machine Learning : An Introduction,» 24 09 2016. [Online]. Available: https://blog.sujeetjaiswal.com/machine-learning-an-introduction-de88d85ebc5d.

[12] F. P. Ron Kohavi, «Glossary of Terms Special Issue on Applications of Machine Learning and the Knowledge Discovery Process,» 1198. [Online]. Available: http://robotics.stanford.edu/~ronnyk/glossary.html.

[13] Wikipedia, «Funzionalità (machine learning),» [Online]. Available: https://en.wikipedia.org/wiki/Feature_(machine_learning).

[14] N. Boldrini, «Cos'è il Machine Learning, come funziona e quali sono le sue applicazioni,» digital360, 01 10 2017. [Online]. Available: https://www.ai4business.it/intelligenza-artificiale/machine-learning/machine-learning-cosa-e-applicazioni/.

[15] E. H. Potential, «The State of Enterprise Machine Learning,» *Algorithmia Research,* Ottobre 2018.

[16] S. I. srl, «Machine Learning,» *trend-machine-learning-magazine.*

[17] D. K. Judith Hurwitz, Machine Learning For Dummies, IBM, 2018.

[18] MathWorks, «Introducing Machine Learning,» in *88174_92991v00_machine_learning_section1_ebook.pdf*, MathWorks.

[19] j. bhattacharjee, «Alcune definizioni chiave di apprendimento automatico,» 28 10 2017. [Online]. Available: https://medium.com/technology-nineleaps/some-key-machine-learning-definitions-b524eb6cb48.

[20] NewTechDojo, «List of Machine Learning Algorithms,» 06 03 2018. [Online]. Available: https://www.newtechdojo.com/list-machine-learning-algorithms/.

[21] Wikipedia, «Matrice di confusione,» [Online]. Available: https://it.wikipedia.org/wiki/Matrice_di_confusione.

[22] Wikipedia, «Training_set,» [Online]. Available: https://it.wikipedia.org/wiki/Training_set.

[23] developers, «Machine Learning Glossary,» 10 2018. [Online]. Available: https://developers.google.com/machine-learning/glossary/#o.

[24] Quora, «What-is-the-difference-between-supervised-and-unsupervised-learning-algorithms,» [Online]. Available: https://www.quora.com/What-is-the-difference-between-supervised-and-unsupervised-learning-algorithms.

[25] S. Favaro, «Che Cos'è Il Machine Learning,» [Online]. Available: http://www.simonefavaro.it/2017/04/07/introduzione-machine-learning/.

[26] joachims, A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization Proceedings of the 14th International Conference on Machine Learning, 1997.

[27] l. F. Angela, Tecniche di Estrazione e Selezione di Caratteristiche per la Classificazione di pagine HTML, 1998/1999.

[28] A. Rezzani, «Naive bayes classifiers,» Data skills s.r.l, [Online]. Available: https://www.dataskills.it/white-paper/naive-bayes-classifiers/.

[29] S. Raschka, Machine Learning con Python, APOGEO.

[30] D. Madushan, «Introduction to K-means Clustering,» 01 12 2017. [Online]. Available: https://medium.com/@dilekamadushan/introduction-to-k-means-clustering-7c0ebc997e00.

[31] kindsonthegenius, «What is K-Means in Clustering in Machine Learning?,» 20 12 2017. [Online]. Available: http://kindsonthegenius.blogspot.com/2017/12/what-is-k-means-in-clustering.html.

[32] U. S. V. M. a. f. e. (. w. code), «Understanding Support Vector Machine algorithm from examples (along with code),» ANALYTICS VIDHYA, 13 09 2017. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/.

[33] A. V. C. TEAM, «A Complete Tutorial on Tree Based Modeling from Scratch (in R & Python),» ANALYTICS VIDHYA, 12 04 2016. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/.

[34] L. M. 8, «Mining Frequent Itemsets – Apriori Algorithm,» http://software.ucv.ro/, [Online]. Available: http://software.ucv.ro/~cmihaescu/ro/teaching/AIR/docs/Lab8-Apriori.pdf.

[35] «APRIORI ALGORITHM,» International School of Engineering , 19 06 2014. [Online]. Available: https://www.slideshare.net/INSOFE/apriori-algorithm-36054672.

[36] ujjwalkarn, «A Quick Introduction to Neural Networks,» the data science blog, 09 08 2016. [Online]. Available: https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/.

[37] L. Govoni, «Rete Neurale, Deep Learning e principali applicazioni,» [Online]. Available: https://lorenzogovoni.com/deep-learning-e-applicazioni/.

[38] ADL, «Un'introduzione al Q-Learning: apprendimento di rinforzo,» freeCodeCamp, [Online]. Available: https://medium.freecodecamp.org/an-introduction-to-q-learning-reinforcement-learning-14ac0b4493cc.

[39] «Internet of Things: servizi migliori e business efficiente,» 13 09 2018. [Online]. Available: https://www.theinnovationgroup.it/blog/internet-of-things-servizi-migliori-e-business-efficiente/?lang=it.

[40] K. B. Mohan S, «A survey on IoT platforms,» 2016.

[41] I. C. Publication, «A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment,» 29 Oct 2018. [Online]. Available: https://ieeexplore.ieee.org/document/7043637.

[42] C. Bellini, «http://automazione-plus.it/wp-content/uploads/sites/3/2016/09/Report-IoT-2015-FINAL.pdf,» 2016.

[43] E. V. Camilla Bellini, «https://www.theinnovationgroup.it/wp-content/uploads/2014/07/TIGPaper_I-trend-evolutivi-dellIoE.pdf,» 2014.

[44] EY, «Internet of Things Human-machine interactions that unlock possibilities,» [Online]. Available: https://www.ey.com/Publication/vwLUAssets/ey-m-e-internet-of-things/$FILE/ey-m-e-internet-of-things.pdf.

[45] IQUII, «IOT: UN MERCATO DESTINATO AD ESPLODERE E RAGGIUNGERE I 3MILA MILIARDI DI DOLLARI ENTRO IL 2020,» 2017. [Online]. Available: https://iquii.com/2017/12/04/iot-mercato-3mila-miliardi-dollari-2020/.

[46] M. T. J. S. R. J.-G. M. L. A. A. H. P. Milan Zdravković, «Survey of Internet-of-Things platforms,» 2016. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01298141/document.

[47] Gartner, «Machina Research IoT Predictions 2017,» 2017. [Online]. Available: https://machinaresearch.com/news/machina-research-iot-predictions-2017.

[48] F. Fabbri, «Internet of Things, 29 miliardi di device connessi nel 2022,» 2017. [Online]. Available: https://www.key4biz.it/internet-of-things-29-miliardi-device-connessi-nel-2022/192930/.

[49] BitMat, «Lo stato dell'arte: il mercato Internet of Things 2017,» 2017. [Online]. Available: https://www.bitmat.it/blog/news/67358/lo-dellarte-mercato-internet-of-things-2017.

[50] A. Frollà, «IoT, il mercato italiano vale quasi 3 miliardi: +40% di crescita in un solo anno,» 2017. [Online]. Available: https://www.corrierecomunicazioni.it/industria-4-0/iot-il-mercato-italiano-vola-a-quota-28-miliardi-40-in-anno/.

[51] L. Zanotti, «IoT, Data Analytics e AI: i nuovi business della comunicazione bidirezionale,» 2017. [Online]. Available: https://www.internet4things.it/open-innovation/iot-data-analytics-e-ai-i-nuovi-business-della-comunicazione-bidirezionale/.

[52] IEEE, «A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment,» 2014. [Online]. Available: https://ieeexplore.ieee.org/document/7043637.

[53] T. I. Corporate. [Online]. Available: https://www.telecomitalia.com/tit/it.html.

[54] N. S. Gill, «Data Ingestion, Processing and Architecture layers for Big Data and IoT,» 2017. [Online]. Available: https://www.xenonstack.com/blog/big-data-engineering/ingestion-processing-big-data-iot-stream/.

[55] TEORIASDADENNY.COM, «Fine Big Data Architecture Framework On Regarding Ingestion Processing And Layers XenonStack,» [Online]. Available: http://teoriasdadenny.com/contemporary-big-data-architecture-framework/fine-big-data-architecture-framework-on-regarding-ingestion-processing-and-layers-xenonstack/.

[56] A. KAFKA, «kafka - Introduction,» [Online]. Available: https://kafka.apache.org/intro.

[57] S. D., «Data Ingest Self Service and Management using Nifi and Kafka,» 2017. [Online]. Available: https://www.slideshare.net/Hadoop_Summit/data-ingest-self-service-and-management-using-nifi-and-kafka.

[58] K. J., «Putting Apache Kafka To Use: A Practical Guide to Building an Event Streaming Platform (Part 1),» 2015. [Online]. Available: https://www.confluent.io/blog/stream-data-platform-1/.

[59] A. Schildkrout, «Kafka Summit. In: Kafka Summit,» 2016. [Online]. Available: https://kafka-summit.org/speakers/aaron-schildkrout/.

[60] C. Yang, «Rheos,» 2017. [Online]. Available: http://www.ebaytechblog.com/2017/03/14/rheos/.

[61] N. T. Blog, «Evolution of the Netflix Data Pipeline,» 2016. [Online]. Available: https://medium.com/netflix-techblog/evolution-of-the-netflix-data-pipeline-da246ca36905.

[62] J. C., «Billions of Messages a Day - Yelp's Real-time Data Pipeline,» 2016. [Online]. Available: https://engineeringblog.yelp.com/2016/07/billions-of-messages-a-day-yelps-real-time-data-pipeline.html.

[63] S. R. O'Neil C, «Doing Data Science: Straight Talk from the Frontline,» "O'Reilly Media, Inc.", 2013.

[64] Wikipedia, «Data analysis,» [Online]. Available: https://en.wikipedia.org/wiki/Data_analysis#Data_processing.

[65] Wikipedia, «Data processing,» [Online]. Available: https://en.wikipedia.org/wiki/Data_processing.

[66] A. Spark, «Apache SparkTM - Unified Analytics Engine for Big Data,» [Online]. Available: https://spark.apache.org/.

[67] A. Spark, «Overview - Spark 2.3.1 Documentation,» [Online]. Available: https://spark.apache.org/docs/latest/.

[68] A. Hadoop, «Welcome to ApacheTM Hadoop®!,» [Online]. Available: http://hadoop.apache.org/.

[69] Wikipedia, «Apache Hadoop - Wikipedia,» [Online]. Available: https://it.wikipedia.org/wiki/Apache_Hadoop.

[70] «Spark-Cassandra Vs Spark-Elasticsearch. In: Stack Overflow,» [Online]. Available: https://stackoverflow.com/questions/32279604/spark-cassandra-vs-spark-elasticsearch.

[71] quora, «Why shouldn't I use ElasticSearch as my primary datastore?,» [Online]. Available: https://www.quora.com/Why-shouldnt-I-use-ElasticSearch-as-my-primary-datastore.

[72] stackoverflow, «Elasticsearch vs Cassandra vs Elasticsearch with Cassandra,» [Online]. Available: https://stackoverflow.com/questions/27054954/elasticsearch-vs-cassandra-vs-elasticsearch-with-cassandra.

[73] G. B. Perez F, «IPython: A System for Interactive Scientific Computing.,» Comput Sci Eng., 2007, p. 9: 21–29..

[74] R. Johnson, «Mesos at Yelp: Building a production ready PaaS,» [Online]. Available: https://events.static.linuxfound.org/sites/events/files/slides/paasta_mesoscon.pdf.

[75] A. Developer, «Mobile Device Management Protocol Reference,» [Online]. Available: https://developer.apple.com/library/content/documentation/Miscellaneous/Reference/MobileDeviceManagementProtocolRef/3-MDM_Protocol/MDM_Protocol.html.

[76] E. A. A. I. E. Community, «Architecture,» [Online]. Available: https://iot.engineersaustralia.org.au/wiki-index/architecture-r6/.

[77] A. Bassi, «La giungla delle architetture di riferimento per l'IoT,» IoT Italy, [Online]. Available: http://www.iotitaly.net/la-giungla-delle-architetture-riferimento-iot/.

[78] K. -. I. o. Things, «L'architettura dell'Internet of Things,» 2016. [Online]. Available: http://www.keplerotech.com/2016/01/08/architettura-internet-of-things/.

[79] P. FREMANTLE, «A REFERENCE ARCHITECTURE FOR THE INTERNET OF THINGS,» WSO2, 2015. [Online]. Available: https://wso2.com/download/getfile/wso2_whitepaper_a-reference-architecture-for-the-internet-of-things.pdf.

[80] ZeroUno, «Le caratteristiche di una piattaforma Internet of Things: perché è importante l'edge Computing,» [Online]. Available: https://www.zerounoweb.it/mobility/internet-of-things-serve-una-piattaforma-basata-sulledge-computing/.

[81] L. J., «GETTING STARTED: WRITING YOUR FIRST PLAYBOOK,» 2017. [Online]. Available: https://www.ansible.com/blog/getting-started-writing-your-first-playbook.

[82] github, «ansible-examples,» [Online]. Available: https://github.com/ansible/ansible-examples.

[83] U. B. M. F. P. F. O. K. F. L. L. R. Jasmin Guth, «A Detailed Analysis of IoT Platform Architectures: Concepts, Similarities, and Differences,» [Online]. Available: http://www.iaas.uni-stuttgart.de/RUS-

data/INBOOK-2018-01%20-%20A%20Detailed%20Analysis%20of%20IoT%20Platform%20Architectures%20-%20Concepts%2C%20Similarities%2C%20and%20Differences.pdf.

[84] A. W. Services, «Applicazioni e soluzioni IoT | Cos'è l'Internet of Things (IoT)?,» [Online]. Available: https://aws.amazon.com/it/iot/.

[85] A. W. Services, «Amazon FreeRTOS– Sistema operativo IoT per microcontroller,» [Online]. Available: https://aws.amazon.com/it/freertos/.

[86] A. W. Services, «AWS Greengrass – Amazon Web Services.,» [Online]. Available: https://aws.amazon.com/it/greengrass/.

[87] A. W. Services, «Panoramica di AWS IoT Core,» [Online]. Available: Available: https://aws.amazon.com/it/iot-core/.

[88] A. W. Services, «IoT Device Management | Onboarding, organizzazione e aggiornamento da remoto,» [Online]. Available: https://aws.amazon.com/it/iot-device-management/.

[89] A. W. Services, «Sicurezza IoT | Gestione della sicurezza dei dispositivi IoT | AWS IoT Device Defender,» [Online]. Available: https://aws.amazon.com/it/iot-device-defender/.

[90] A. W. Services, «Panoramica di AWS IoT Analytics,» [Online]. Available: https://aws.amazon.com/it/iot-analytics/.

[91] A. W. Services, «Panoramica su AWS IoT 1-Click – Creazione, con un solo clic, di un trigger AWS Lambda per qualsiasi dispositivo.,» [Online]. Available: https://aws.amazon.com/it/iot-1-click/.

[92] A. W. Services, «Panoramica di AWS IoT Button – Dash Button programmabile nel cloud.,» [Online]. Available: https://aws.amazon.com/it/iotbutton/.

[93] A. Greengrass, «What Is AWS IoT Greengrass?,» [Online]. Available: https://docs.aws.amazon.com/greengrass/latest/developerguide/what-is-gg.html.

[94] F. Futuro, «PTC consolida competenze nell'IoT con Axeda,» Fabbrica Futuro, [Online]. Available: https://www.fabbricafuturo.it/ptc-consolida-competenze-nelliot-axeda/.

[95] D. W. F. W. Marcus Köhler, «Platforms for the Internet of Things – An Analysis of Existing Solutions,» [Online]. Available: http://cocoa.ethz.ch/downloads/2014/02/1682_20140212%20-%20Bocse.pdf.

[96] A. W. Services, «AWS Greengrass – Run AWS Lambda Functions on Connected Devices,» 2017. [Online]. Available: https://aws.amazon.com/blogs/aws/aws-greengrass-run-aws-lambda-functions-on-connected-devices/.

[97] B. Labs, «Kevin Barr of Pinkston Digital,» [Online]. Available: https://buglabs.net/about.

[98] dweet.io, «Share your thing- like it ain't no thang,» [Online]. Available: http://dweet.io/faq.

[99] freeboard, «Dashboards For the Internet Of Things,» [Online]. Available: https://freeboard.io/.

[100] E. I. S. P. Platform, «Platform data sheet,» [Online]. Available: https://evrythng.com/resources/data-sheets/platform-data-sheet/.

[101] E. I. S. P. Platform, «Features,» 2015, [Online]. Available: https://evrythng.com/platform/features/.

[102] E. I. S. P. Platform, «Developer Tools,» 2015, [Online]. Available: https://evrythng.com/platform/developer-tools/.

[103] G. Cloud, «Google Cloud IoT,» [Online]. Available: https://cloud.google.com/solutions/iot/.

[104] G. Cloud, «Cloud IoT Core,» [Online]. Available: https://cloud.google.com/iot-core/.

[105] Rahul., «Top 20 IoT Platforms in 2018,» Internet Of Things Wiki, 2017. [Online]. Available: https://internetofthingswiki.com/top-20-iot-platforms/634/.

[106] IBM, «IBM Watson IoT Platform - Panoramica - Italia,» [Online]. Available: https://www.ibm.com/it-it/marketplace/internet-of-things-cloud.

[107] K. I. platform, «Technical capabilities and IoT architecture of Kaa,» [Online]. Available: https://www.kaaproject.org/.

[108] M. Azure, «Website,» [Online]. Available: https://docs.microsoft.com/it-it/azure/iot-hub/about-iot-hub.

[109] OpenMTC, «OpenMTC in a nutshell,» [Online]. Available: https://www.openmtc.org/.

[110] OpenMTC, «OpenMTC Integrations,» [Online]. Available: https://www.openmtc.org/integrations.html.

[111] Oracle, «Oracle Internet of Things Cloud Service,» [Online]. Available: http://www.oracle.com/us/solutions/internetofthings/iot-cloud-service-ds-3209769.pdf.

[112] SensorCloud, «What is SensorCloud?,» LORD Sensing , [Online]. Available: https://www.sensorcloud.com/documentation.

[113] SiteWhere, «SiteWhere Platform,» [Online]. Available: http://sitewhere.io/docs/2.0.0-rc2/platform/.

[114] SiteWhere, «SiteWhere Architecture,» [Online]. Available: http://sitewhere.io/docs/2.0.0-rc2/platform/architecture.html.

[115] SiteWhere, «Microservices Overview,» [Online]. Available: http://sitewhere.io/docs/2.0.0-rc2/platform/microservice-overview.html.

[116] S. C. D. Documentation, «Overview,» [Online]. Available: https://docs.smartthings.com/en/latest/getting-started/overview.html.

[117] S. C. D. Documentation, «Architecture,» [Online]. Available: https://docs.smartthings.com/en/latest/architecture/index.html.

[118] T. INSTRUMENTS, «Internet of Things - IoT Partners,» TI.com, [Online]. Available: http://www.ti.com/ww/en/internet_of_things/Thingsquare-for-TI-IoT-development-kits.html#.

[119] SearchERP, «What is PTC ThingWorx? - Definition from WhatIs.com,» [Online]. Available: https://searcherp.techtarget.com/definition/PTC-ThingWorx.

[120] Xively, «IoT Platform for Connected Devices,» [Online]. Available: https://xively.com/.

[121] Google, «teachablemachine,» [Online]. Available: https://teachablemachine.withgoogle.com/.

[122] A. Konrad, «https://www.forbes.com/sites/alexkonrad/2017/05/31/mary-meeker-internet-trends-for-2017/#5d993091143e,» Forbes, 31 05 2017. [Online]. Available: Here Is Mary Meeker's Internet Trends Report For 2017.

[123] wikipedia, «Sistema_di_raccomandazione,» [Online]. Available: https://it.wikipedia.org/wiki/Sistema_di_raccomandazione.

[124] C. Pinela, «Recommender Systems — User-Based and Item-Based Collaborative Filtering,» Medium, 06 11 2017. [Online]. Available: https://medium.com/@cfpinela/recommender-systems-user-based-and-item-based-collaborative-filtering-5d5f375a127f.

[125] AI4Business, «Riconoscimento facciale: che cos'è e perché sta rivoluzionando smartphone, sicurezza e sistemi di pagamento,» 22 11 2017. [Online]. Available: https://www.ai4business.it/sicurezza/riconoscimento-facciale/.

[126] «Tra contesti applicativi e standard,» Soiel International Srl , 16 07 2015. [Online]. Available: https://www.soiel.it/news/dettaglio/tra-contesti-applicativi-e-standard/#.

[127] E. i. C. I. Eric Knorr, «How PayPal beats the bad guys with machine learning,» InfoWorld, 13 04 2015. [Online]. Available: https://www.infoworld.com/article/2907877/machine-learning/how-paypal-reduces-fraud-with-machine-learning.html.

[128] Ravelin, «Apprendimento automatico e prevenzione delle frodi,» [Online]. Available: https://www.ravelin.com/fraud-guides/machine-learning-and-fraud-prevention.

[129] E. Vidra, «30 startup di Machine Intelligence da tenere d'occhio in Israele,» Medium, 16 02 2017.

[Online]. Available: https://medium.com/@ediggs/30-machine-intelligence-startups-to-watch-in-israel-a05b6597c4a5.

[130] hubspot, «In a not-so-distant future, there's a bleak, forsaken landscape.,» [Online]. Available: https://www.hubspot.com/stories/chatbot-marketing-future.

[131] gartner, «The Road to Enterprice AI,» [Online]. Available: https://www.gartner.com/imagesrv/media-products/pdf/rage_frameworks/rage-frameworks-1-34JHQ0K.pdf.

[132] gartner, «Gartner Predicts a Virtual World of Exponential Change,» 18 10 2016. [Online]. Available: https://www.gartner.com/smarterwithgartner/gartner-predicts-a-virtual-world-of-exponential-change/.

[133] codecademy, «What is Scikit-Learn,» [Online]. Available: https://www.codecademy.com/articles/scikit-learn.

[134] X. Su, Introduction to Big Data, Forfatter og Stiftelsen TISIP.

[135] https://datafloq.com. [Online].

[136] G. Press, «6 Predictions For The $203 Billion Big Data Analytics Market,» 20 Jan 2017. [Online]. Available: https://www.forbes.com/sites/gilpress/2017/01/20/6-predictions-for-the-203-billion-big-data-analytics-market/#6b1401062083.

[137] R. v. d. Meulen, «2017: The Year That Data and Analytics Go Mainstream,» 24 01 2017. [Online]. Available: https://www.gartner.com/smarterwithgartner/2017-the-year-that-data-and-analytics-go-mainstream/.

[138] L. Columbus, «Roundup Of Analytics, Big Data & BI Forecasts And Market Estimates, 2016,» 20 08 2016. [Online]. Available: https://www.forbes.com/sites/louiscolumbus/2016/08/20/roundup-of-analytics-big-data-bi-forecasts-and-market-estimates-2016/#4928ef066f21.

[139] Quora, «https://www.quora.com/What-is-the-difference-between-a-data-scientist-and-a-machine-learning-engineer?redirected_qid=13377431,» [Online].

[140] IntelligenzaArtificiale.it, «Data Mining,» [Online]. Available: http://www.intelligenzaartificiale.it/data-mining/.

[141] B. Marr, «How Is Big Data Used In Practice? 10 Use Cases Everyone Must Read,» Forbes, [Online]. Available: https://www.bernardmarr.com/default.asp?contentID=1076.

[142] dezyre, «5 Big Data Use Cases- How Companies Use Big Data,» 2015. [Online]. Available: https://www.dezyre.com/article/5-big-data-use-cases-how-companies-use-big-data/155.

[143] kdnuggets, «Big Data and Data Science for Security and Fraud Detection,» [Online]. Available: https://www.kdnuggets.com/2015/12/big-data-science-security-fraud-detection.html.

[144] A. Microsoft, «What-is-cloud-computing,» [Online]. Available: https://azure.microsoft.com/it-it/overview/what-is-cloud-computing/.

[145] «NIST SP,» pp. 800-145, Sept. 2011.

[146] smechannel, «PaaS, SaaS, IaaS: The forefront of Digital India,» 2016. [Online]. Available: http://www.smechannels.com/paas-saas-iaas-the-forefront-of-digital-india/.

[147] D. Chou. [Online]. Available: https://dachou.github.io/2018/09/28/cloud-service-models.html.

[148] National Institute of Standards and Technology, «NIST,» [Online]. Available: https://www.nist.gov/.

[149] P. B. R. C. a. N. D. I. P. C. M. Satyanarayanan, «"The case for VM-based cloudlets in mobile computing",» pp. Vol. 8, no. 4, pp. 14–23, , Oct./Dec. 2009.

[150] W. e. a. Shi, «Edge computing: Vision and challenges,» pp. 637-646, (2016).

[151] O. C. A. W. Group, «OpenFog Architecture Overview - White Paper,» 7 Dec. 2016. [Online]. Available: http://www.openfogconsortium.org/wp-content/uploads/OpenFog-Architecture-

Overview-WP-2-2016.pdf.

[152] M. &. Z. T. Chiang, « Fog and IoT: An overview of research opportunities,» IEEE Internet of Things Journal, 2016, pp. 854-864.

[153] CiTRIX, «What is multi-cloud?,» [Online]. Available: https://www.citrix.it/glossary/what-is-multi-cloud.html .

[154] C. H. Q. L. Z. L. K. &. H. F. Yang, «Big Data and cloud computing: innovation opportunities and challenges,» International Journal of Digital Earth, 2017, pp. 10(1), 13-53..

[155] A. D. D. W. P. V. &. P. A. Botta, «Integration of cloud computing and internet of things: a survey.,» Future Generation Computer Systems, 2016, pp. 56, 684-700..

[156] D. Pop, «Machine learning and cloud computing: Survey of distributed and saas solutions,» arXiv preprint arXiv:1603.08767., 2016.

[157] P. L. J. H. Z. L. T. G. C. Z. Y. S. M. &. C. K. Li, «Multi-key privacy-preserving deep learning in cloud computing.,» Future Generation Computer Systems, 2017, pp. 74, 76-85..

[158] A. &. N. N. J. Vakili, «Comprehensive and systematic review of the service composition mechanisms in the cloud environments.,» Journal of Network and Computer Applications, 2017, pp. 81, 24-36..

[159] M. H. Z. M. &. H. C. T. Ghahramani, «Toward cloud computing QoS architecture: Analysis of cloud systems and cloud services.,» IEEE/CAA Journal of Automatica Sinica, 2017, pp. 4(1), 6-18..

[160] Y. M. A. A. M. H. &. S. W. Yu, Cloud computing security and privacy: Standards and regulations., 2017.

[161] A. &. V. R. (. Suresh, «Competent resource provisioning and distribution techniques for cloud computing environment,» Cluster Computing, pp. 1-8.

[162] C. P. Laurence Goasduff, «Gartner Says Worldwide Public Cloud Services Market to Grow 18 Percent in 2017,» Gartner , 22 02 2017. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2017-02-22-gartner-says-worldwide-public-cloud-services-market-to-grow-18-percent-in-2017.

[163] Cloudify, «2017:State of Enterprise Multi-Cloud,» 2017. [Online]. Available: http://wp.cloudify.co/wp-content/uploads/2017/09/2017-State-of-Enterprise-Multi-Cloud-Report.pdf.

[164] McAfee, «Navigating a Cloudy Sky: Practical Guidance and the State of Cloud Security,» [Online]. Available: https://www.mcafee.com/us/solutions/lp/cloud-security-report.html.

[165] R. D. V. E. D. S. e. F. M. D. T. Di Noia, «Semantic Web - Tra ontologie e Open Data,» Apogeo, 2013.

[166] Wikipedia, «Semantic Web,» [Online]. Available: https://en.wikipedia.org/wiki/Semantic_Web.

[167] Wikipedia, «Web of Things,» [Online]. Available: https://en.wikipedia.org/wiki/Web_of_Things.

[168] C. H. e. S. S. A. Sheth, «Semantic Sensor Web,» IEEE Internet computing, 2008, pp. vol. 12, n. 4.

[169] A. K. e. C. Jennings, «Sensor Measurement Lists (SenML),» August 2018. [Online]. Available: https://tools.ietf.org/html/rfc8428..

[170] P. B. L. B. R. G.-C. O. C. S. C. J. G. M. H. C. H. A. H. e. o. M. Compton, «The SSN ontology of the W3C semantic sensor network incubator group,,» Web semantics: science, services and agents on the World Wide Web, 2012, pp. Vol. 17, pp 25-32.

[171] w3c, «Semantic Sensor Network Ontology,» w3c, [Online]. Available: https://w3c.github.io/sdw/ssn/.

[172] w3, «Semantic Sensor Network Ontology,» October 2017. [Online]. Available: https://www.w3.org/TR/vocab-ssn/#intro..

[173] Kubernetes, «Kubernetes.png,» [Online]. Available: https://commons.wikimedia.org/w/index.php?curid=53571935.

[174] Microsoft, «Mesosphere DC/OS Open Source - Panoramica,» [Online]. Available: https://azuremarketplace.microsoft.com/it-it/marketplace/apps/mesosphere.dcos.

[175] Docker, «What is a Container,» [Online]. Available: https://www.docker.com/resources/what-container.

[176] Wikipedia, «ArquiteturaDocker,» [Online]. Available: https://commons.wikimedia.org/wiki/File:ArquiteturaDocker.png.

[177] «Apagche spark,» [Online]. Available: https://spark.apache.org/.

[178] M. Azure, «Data Lake,» [Online]. Available: https://azure.microsoft.com/en-us/solutions/data-lake/.

[179] AWS, «Data Lake su AWS,» [Online]. Available: https://aws.amazon.com/it/answers/big-data/data-lake-solution/.

[180] Huawei, «Data Lake Factory,» [Online]. Available: https://www.huaweicloud.com/en-us/product/dlf.html.

[181] D. Ramel, «High Demand and Salaries for Apache Cassandra, Other NoSQL Skills,» 2016, [Online]. Available: https://adtmag.com/articles/2016/03/15/cassandra-nosql-skills.aspx.

[182] Apache, «What is Cassandra?,» [Online]. Available: http://cassandra.apache.org/.

[183] Apache, «Welcome to Apache HBase™,» [Online]. Available: https://hbase.apache.org/.

[184] data-flair, «Applications of HBase,» [Online]. Available: https://data-flair.training/blogs/hbase-use-cases/.

[185] basho, «Document Databases Explained,» [Online]. Available: http://basho.com/resources/document-databases/.

[186] I. MongoDB, «MongoDB,» [Online]. Available: https://www.mongodb.com.

[187] data-flair, «Vantaggi di MongoDB | Svantaggi di MongoDB,» [Online]. Available: https://data-flair.training/blogs/advantages-of-mongodb/.

[188] DB-ENGINES, «DB-Engines Ranking,» [Online]. Available: https://db-engines.com/en/ranking.

[189] I. Neo4j, «What is Neo4j?,» [Online]. Available: https://neo4j.com/.

[190] RubyGarage, «Capabilities of the Neo4j Graph Database with Real-life Examples,» [Online]. Available: https://rubygarage.org/blog/neo4j-database-guide-with-use-cases.

[191] qlik, «qlik,» [Online]. Available: https://www.qlik.com/us/.

[192] TechTarget, «Data visualization,» [Online]. Available: https://searchbusinessanalytics.techtarget.com/definition/data-visualization.

[193] E. Burns, «Qlik Sense vs. QlikView: How the two Qlik tools compare,» TechTarget, [Online]. Available: https://searchbusinessanalytics.techtarget.com/feature/Qlik-Sense-vs-QlikView-How-the-two-Qlik-tools-compare.

[194] T. Technology. [Online]. Available: https://www.tableau.com/products/technology.

[195] S. Lab, «Mini-ME: il Mini Matchmaking Engine,» [Online]. Available: http://sisinflab.poliba.it/swottools/minime/.

[196] «Mini-ME Matchmaker and Reasoner for the Semantic Web of Things,» [Online]. Available: https://www.igi-global.com/chapter/mini-me-matchmaker-and-reasoner-for-the-semantic-web-of-things/196442.

[197] N. A. E. D. K. T. C. Miltiadis D. Lytras, *Innovations,Developments, and Applications of Semantic Web and Information Systems,* IGI Global.

[198] OSF, «Open Semantic Framework,» [Online]. Available: http://opensemanticframework.org/.

[199] T. A. S. Foundation, «Apache Jena,» [Online]. Available: https://jena.apache.org/.

[200] Oracle, «Oracle Spatial and Graph 18c RDF,» [Online]. Available:

https://download.oracle.com/otndocs/products/spatial/pdf/SG_RDF_Knowledge_Graph_18c_Data_Sheet.pdf.

[201] wikipedia, «Webinos,» [Online]. Available: https://en.wikipedia.org/wiki/Webinos.

[202] CENSIS, «Il capitolo «Comunicazione e media» del 46° Rapporto Censis sulla situazione sociale del Paese/2012,» [Online]. Available: http://www.censis.it/7?shadow_comunicato_stampa=118937.

[203] Davide.it, «Il fenomeno dei Social Network,» [Online]. Available: http://www.davide.it/whitepaper/2010/giugno/social_network.pdf.

[204] HTML.it, «Graph API,» [Online]. Available: http://www.html.it/pag/18898/le-graph-api/.

[205] Facebook Developers, «Become a Facebook Developer,» [Online]. Available: http://developers.facebook.com.

[206] Facebook API, «The Graph API,» [Online]. Available: https://developers.facebook.com/docs/graph-api/.

[207] Facebook API User, «Graph API: User,» [Online]. Available: https://developers.facebook.com/docs/graph-api/reference/user/.

[208] Twitter Developers, «Il portale di Twitter per developers,» [Online]. Available: https://dev.twitter.com/.

[209] Twitter API, «API per Twitter,» [Online]. Available: https://dev.twitter.com/docs/api.

[210] Linkedin API, «API per Linkedin,» [Online]. Available: http://developer.linkedin.com/apis.

[211] YouTube API, «YouTube API v2.0,» [Online]. Available: https://developers.google.com/youtube/2.0/developers_guide_protocol.

[212] YouTube Developers, «YouTube Developers Guide,» [Online]. Available: https://developers.google.com/youtube/.

[213] Flickr API, «Servizi Flickr,» [Online]. Available: http://www.flickr.com/services/api/.

[214] Viadeo Graph API, «API per Viadeo,» [Online]. Available: http://dev.viadeo.com/documentation.

[215] Google+ API, «Google Developers,» [Online]. Available: https://developers.google.com/+/api/ .

[216] Google APIs Client Library for Python, «google-api-python-client,» [Online]. Available: http://code.google.com/p/google-api-python-client/.

[217] Google APIs Client Library for Java, «google-api-java-client,» [Online]. Available: http://code.google.com/p/google-api-java-client/.

[218] Google API Client, «google-api-ruby-client,» [Online]. Available: http://code.google.com/p/google-api-ruby-client/ .

[219] Google People, «Google Developers,» [Online]. Available: https://developers.google.com/+/api/latest/people .

[220] Apigee console, «Twitter API console,» [Online]. Available: https://apigee.com/console.

[221] A. M. e. al., Maximum Entropy Markov Models for Information Extraction and Segmentation, 17th ICML, 2000.

[222] V. L. a. R. M. J. Jeon, Automatic Image Annotation and Retrieval using CrossMedia Relevance Models, Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, p. 119-126, 2003.

[223] H. H. Lawrence Reeve, Survey of Semantic Annotation Platforms, Lawrence Reeve, Hyoil Han, Survey ofSAC '05 Proceedings of the 2005 ACM symposium on Applied computing, p. 1634-1638, 2005.

[224] Semantic Turkey, «Semantic Turkey (software),» [Online]. Available: http://semanticturkey.uniroma2.it/.

[225] M.-R. K. E. P. a. R. R. S. Jose Kahan, Annotea: An Open RDF Infrastructure for Shared Web

Annotations, Hong Kong, China: In Proceedings of the Tenth Inter-national World Wide Web Conference, pp. 623-632, May, 2001.

[226] W. H. Paul Kogut, «AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages,» First International Conference on Knowledge Capture (K-CAP 2001) - Workshop on Knowledge Markup and Semantic Annotation, 2001. [Online]. Available: http://km.aifb.kit.edu/ws/semannot2001/positionpapers/AeroDAML3.pdf.

[227] Armadillo, «Armadillo (software),» [Online]. Available: http://www.aktors.org/technologies/armadillo/.

[228] B. N. F. C. Sam Chapman, «Armadillo: Integrating Knowledge for the Semantic Web,» [Online]. Available: http://kushmerick.org/nick/research/Dagstuhl-MLSW/proceedings/chapman-norton-ciravegna.pdf.

[229] B. P. I. T. D. M. a. D. O. Atanas Kiryakov, Semantic Annotation, Indexing, and Retrieval, Journal of Web Semantics, 2(1):49-79, December 2004.

[230] GATE, «GATE (software),» [Online]. Available: http://gate.ac.uk/.

[231] E. M. J. D. M. L. A. S. a. F. C. Maria Vargas-Vera, MnM: Ontology Driven Tool for Semantic Markup, In Proceedings of the Workshop Semantic Authoring, Annotation & Knowledge Markup (SAAKM 2002) with ECAI .

[232] D. M. a. V. T. a. K. B. a. H. C. a. Y. Wilks, «MUSE: a MUlti-Source Entity recognition system,» 2003. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.2.8278&rep=rep1&type=pdf.

[233] S. S. a. F. C. Siegfried Handschuh, S-CREAM - Semi-automatic CREAtion of Metadata, Madrid, Spain: In Proceedings of the European Conference on Knowledge Acquisition and Management (EKAW-2002), October, 2002.

[234] N. E. D. G. D. G. R. G. A. J. T. K. K. S. M. S. R. A. T. J. A. T. a. J. Y. Z. Stephen Dill, A Case for Automated Large Scale Semantic Annotations, Journal of Web Semantics, 1(1):115-132, December 2000.

[235] Apolda, «Automated Processing of Ontologies with Lexical Denotations for Annotation,» [Online]. Available: http://apolda.sourceforge.net/.

[236] Jena, «Apache Jena,» [Online]. Available: http://jena.apache.org/.

[237] Sesame, «Sesame (software),» [Online]. Available: http://www.openrdf.org/, https://bitbucket.org/openrdf/sesame.

[238] Virtuoso, «Virtuoso OpenSource (software),» [Online]. Available: http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/.

[239] 4store, «4store (software),» [Online]. Available: http://4store.org/.

[240] OWLIM, «OWLIM (software),» [Online]. Available: http://www.ontotext.com/owlim.

[241] D2RQ, «D2RQ (software),» [Online]. Available: http://d2rq.org/.

[242] D. Server, «D2R Server (software),» [Online]. Available: http://d2rq.org/d2r-server.

[243] O. API, «OWL API (software),» [Online]. Available: http://owlapi.sourceforge.net/.

[244] T. Tres, «Tema Tres (software),» [Online]. Available: http://www.vocabularyserver.com/.

[245] UIMA, «Apache UIMA,» [Online]. Available: http://uima.apache.org/.

[246] GATE, «GATE Embedded (software),» [Online]. Available: http://gate.ac.uk/family/embedded.html.

[247] LingPipe, «LingPipe (software),» [Online]. Available: http://alias-i.com/lingpipe/.

[248] MALLET, «MALLET (software),» [Online]. Available: http://mallet.cs.umass.edu/index.php/Main_Page.

[249] WEKA, «WEKA (software),» [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/.

[250] OpenNLP, «Apache OpenNLP (software),» [Online]. Available: http://opennlp.apache.org/.

[251] Mahout, «Apache Mahout (software),» [Online]. Available: http://mahout.apache.org/.

[252] TreeTagger, «TreeTagger (software),» [Online]. Available: http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/.

[253] A. Swartz, «Request for Comments: 3870,» September 2004. [Online]. Available: http://www.ietf.org/rfc/rfc3870.txt.

[254] W3C, «OWL Web Ontology Language – Reference - W3C Recommendation,» 10 February 2004. [Online]. Available: http://www.w3.org/TR/owl-ref/.

[255] S. P. e. al., «RDF/Topic Maps Interoperability Task Force,» 2006. [Online]. Available: http://www.w3.org/2001/sw/BestPractices/RDFTM/.

[256] Ontopia, «Ontopia (software),» [Online]. Available: http://www.ontopia.net/.

[257] MajorToM. [Online]. Available: http://code.google.com/p/majortom/.

[258] tinyTiM. [Online]. Available: http://tinytim.sourceforge.net/.

[259] R. T. Maps. [Online]. Available: http://rtm.topicmapslab.de/.

[260] L. O. D. Italia. [Online]. Available: http://www.linkedopendata.it/.

[261] D. italiana. [Online]. Available: http://it.dbpedia.org/.

[262] D2RQ. [Online]. Available: http://d2rq.org/.

[263] Mokabyte, «Introduzione a D2R Server,» [Online]. Available: http://www2.mokabyte.it/cms/article.run?articleId=PJ9-3UQ-BB1-IQF_7f000001_13046033_4173cf23.

[264] V. Sponger. [Online]. Available: http://virtuoso.openlinksw.com/presentations/Virtuoso_Sponger_1/Virtuoso_Sponger_1.html.

[265] V. U. Server. [Online]. Available: http://virtuoso.openlinksw.com/.

[266] Aperture. [Online]. Available: http://sourceforge.net/apps/trac/aperture/wiki/Overview.

[267] Any23. [Online]. Available: http://any23.apache.org/.

[268] Any23, «Extractors,» [Online]. Available: http://any23.apache.org/extractors.html.

[269] Microformats. [Online]. Available: http://microformats.org/.

[270] Microformat. [Online]. Available: http://microform.at/.

[271] T. F. o. a. F. (. project. [Online]. Available: http://www.foaf-project.org/.

[272] GEO. [Online]. Available: http://microformats.org/wiki/geo.

[273] hRecipe. [Online]. Available: http://microformats.org/wiki/hrecipe.

[274] hReview. [Online]. Available: http://microformats.org/wiki/hreview.

[275] OpenGraph. [Online]. Available: https://developers.facebook.com/docs/opengraph/.

[276] JSON-LD. [Online]. Available: http://json-ld.org/.

[277] M. Lanthaler, «On Using JSON-LD to Create Evolvable RESTful Services,» [Online]. Available: http://www.markus-lanthaler.com/research/on-using-json-ld-to-create-evolvable-restful-services.pdf.

[278] G. Kellogg, «JSON for Linked Data,» [Online]. Available: http://www.slideshare.net/gkellogg1/json-for-linked-data.

[279] SPARQL. [Online]. Available: http://www.w3.org/TR/sparql11-query/.

[280] LODMS. [Online]. Available: https://grips.semantic-web.at/display/public/LDM/Introduction;jsessionid=2581D3305FFF86A3451CB4DEB81E8884.

[281] Liferay. [Online]. Available: http://www.liferay.com/.

[282] Liferay, «Service Builder,» [Online]. Available: http://www.liferay.com/it/documentation/liferay-portal/6.0/development/-/ai/service-build-2.

[283] Hibernate. [Online]. Available: http://www.hibernate.org/.

[284] PostrgreSQL. [Online]. Available: http://www.postgresql.org/.

[285] Jersey. [Online]. Available: https://jersey.java.net/.

[286] Sgvizler. [Online]. Available: https://code.google.com/p/sgvizler/.

[287] S. M. L. a. C. H. O. Rachel M. Heck, «A survey of web annotation systems,» 1999. [Online]. Available: http://www.math.grin.edu/~rebelsky/Blazers/Annotations/Summer1999/Papers/survey_paper.html .